8-2017

A Dataset of Gaze Behavior in VR Faithful to Natural Statistics

Brendan John
bmj8778@rit.edu

A Dataset of Gaze Behavior in VR Faithful to Natural Statistics

by

Brendan John

THESIS

Presented to the Faculty of the Golisano College of Computer and Information Sciences
Rochester Institute of Technology
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science, Computer Science

Rochester Institute of Technology

August 2017

A Dataset of Gaze Behavior in VR Faithful to Natural Statistics

APPROVED BY

SUPERVISING COMMITTEE:

Dr. Reynold Bailey, Supervisor

Dr. Gabriel Diaz, Reader

Dr. Joe Geigel, Observer

Acknowledgments

I wish to thank my committee of Dr. Reynold Bailey, Dr Gabriel Diaz, and Dr. Joe Geigel along with all of the other faculty and students that I've worked with in my time at RIT. This could not have been accomplished without the resources and creative students in the CS Graphics and Applied Perception lab and the PerForm lab. Specifically I'd like to thank Kamran Binaee, Evan Krueger and Rakshit Kothari of the PerForm Lab for their assistance on the project, as well as former graduate student Srinivas Sridharan for mentorship early on in my research career.

I would also like to thank my Mother, Father, and Sister for supporting me on my journey at RIT, and the RIT Future Stewards Program for providing social, career, and academic support over the last 6 years.

Abstract

A Dataset of Gaze Behavior in VR Faithful to Natural Statistics

Brendan John
Rochester Institute of Technology, 2017

Supervisor: Dr. Reynold Bailey

Eye tracking technology is advancing swiftly and many areas of research have begun taking advantage of this. Existing eye trackers project gaze onto a 2D plane, whether it be the display of a head mounted virtual reality (VR) helmet or an image of a real life scene the user is in. This allows us to easily analyze what a viewer is looking at, but limits classification of gaze behaviors from this type of signal. Instead, a system that takes into account head movements within the same space as gaze velocity allows researchers to classify more advanced gaze behaviors such as smooth pursuits and fixations resulting from vestibulo-ocular reflex. For this work data is collected in real world environments where head and gaze movements are recorded over a variety of tasks. The resulting data is then used to construct a distribution of naturally occurring gaze behaviors. This distribution is then used to drive a VR

data collection experiment that elicits specific gaze behaviors such as fixations and saccades with specific velocities and directions. A dataset of 12 subjects was collected while subjects play a shooting game in the virtual world. Data was analyzed to see if the intended eye movements were produced, and also to compare the eye movements that occur in fast versus slow presentation of targets.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

There have been rapid advances in head mounted display technologies, and at the same time the costs for eye tracking integration and head tracking have fallen. This has provided more access for researchers to study eye movements in 3D physical and virtual environments, as opposed to the traditionally restricted lab setting. A major problem in the field is the difference between classification of eye movements collected in a laboratory setting with those of gaze behaviors in a real world setting. There is a large corpus of literature that has studied eye movements in isolation resulting in over 20 different advanced classes for eye movements, which include varieties of nystagmus, square-wave jerks and other noisy eye movements [45]. The majority of these take place in a laboratory where external stimuli is removed and a specific set of instructions are provided. This produces a deep understanding of low level eye movements, which are used to construct models and theories of the structure of the human brain and how it handles visual information [20]. This isolated environment is not one that humans are used to however. These experiments elicit specific movements and allow for in depth analysis of eye movements, however in the real world this analysis is not so easy. Even performing a task as simple as walking down a hallway subjects us to many different stimuli and cues

that influence where and how we look around the environment. While these advanced classes of eye movements may occur during this task, it is nearly impossible to distinguish them from other eye movements without spending lots of time analyzing the collected data. Realizing this difference the goal of researchers studying vision in natural environments is to classify gaze behaviors as opposed to eye movements specifically. To do this head movement data is needed in conjunction with raw eye movements.

Another problem that researchers face with eye movement classification is that the current methods for classifying gaze behavior were developed for desktop eye trackers with 2D displays, which work best with no head movement or 3D pose changes. Traditional event detection algorithms classify based solely on eye-in-world movement, which is limited to identifying events where the head orientation does not have an effect. An example of this can be seen when you stare at your thumb and rotate your head side to side. If you continue staring at your thumb you are maintaining fixation on a point in the world, however your eyes are moving in the opposite direction of your head. From the eye tracker's perspective the eyes are moving from side to side and the data may be labeled incorrectly. This type of eye movement is called vestibulo-ocular reflex (VOR), and can be seen in Figure 1.1.

Work on the detection of gaze events in a 3D setting is still in its infancy, but initial efforts are being made to collect data that accurately reflects eye movements made in a natural environment through the use of mobile eye tracking glasses and head tracking. With the escalation of Virtual Reality

Figure 1.1: An example of angular VOR as the head rotates forward and the eyes remain fixed forward due to a counter rotation. Image from Tasman and Jaeger [36].

(VR) and Augmented Reality (AR) we can also extend eye and head tracking to purely virtual environments in the case of VR, or mixed environments in the case of AR. Data collected in VR can now be collected and compared to 3D gaze data collected in the real world for researchers who are interested in studying perceptual differences in VR, specifically concerning human vision. Since VR gives us ultimate control over what the viewer sees and interacts with one thing we can do is collect naturalistic gaze data in the real world, and try to elicit the same behavior in the virtual world. Through this motivation we decided to collect gaze data with both eye and head signals of humans performing natural tasks in the real world, and derived a distribution of eye movements including direction, velocity and duration of eye movements relative to the head's frame of reference. This distribution can then be input to a VR experiment which attempts to recreate these specific eye movements with the goal of collecting data that matches movements that occur most often in the real world.

This was accomplished through a VR game designed to produce specific gaze behavior through automated presentation of stimuli and interactions for the user. The frequency of specific eye movements that appeared in this experiment were manipulated in such a way that "edge cases" are sampled less than more naturally occurring movements. Using targets that the subject will track we manipulate gaze and recreate the collected eye movements. This is not guaranteed to work with humans, as the coordination between head and eyes may be different than when the eye movement was originally made.

The main goal of this work is to bridge the gap between 2D eye tracking data and 3D VR environments while building a more complete database of 3D gaze behaviors. We take into account data collected in real world environments and measure how reproducible these are in VR. This data can then be used to produce more accurate classification of natural eye movements, thus allowing researchers to build more robust event detection techniques.

Chapter 2

Background

Eye tracking technology is certainly not new, with vision researchers recording eye movements over 80 years into the past [3]. In this time vision researchers have delved deep into the subject of eye movements in an attempt to classify and understand the sometimes small but important low level movements and their biological purpose. Thus, classifications arise that can generally group eye movements into Fixational (FEM), Rapid (REM), and Slow (SEM) or into even more specific sub-categories [20]. The following sections aim to briefly introduce the reader to the human visual system, show that different eye movement classifications are used at different scales, and also highlight the current state of event detection through generalized and machine learning techniques.

2.1 Eye Movements

Humans rely heavily on vision to navigate, extract information, and interpret the environment around us. The retina of the eye responds to light passing through the pupil via rods and cones. A distribution of these rods and cones can be seen in Figure 2.1. As seen in the figure the density of cones

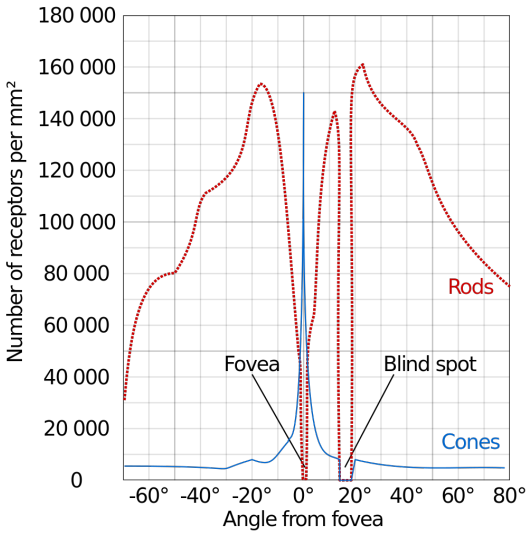

Figure 2.1: Distribution of rods and cones in the retina. At the fovea, which is the center of gaze, the density of cones peak then drop rapidly as you move away. Cones are responsible for high fidelity visual acuity, while rods provide low visual acuity and are sensitive to motion. Thus, visual acuity is best at the center of gaze and much lower in the periphery. Also pictured is the blind spot that exists due to the optical disk, where all retinal signals are routed to the brain. Image courtesy of Cmglee [41].

is the highest in the center of vision (fovea) and quickly drops off after one degree on either side of the fovea. These cones are responsible for high acuity vision which means the ability to distinguish high detail information drops off significantly as you move away from the fovea. At one point in time only a small amount of the scene can be seen in high detail, and to compensate our eyes quickly scan the scene while our brain processes and pieces everything together.

Moments when the fovea is concentrated around a point and gathering high detail information is called a fixation. The eyes are not perfectly steady

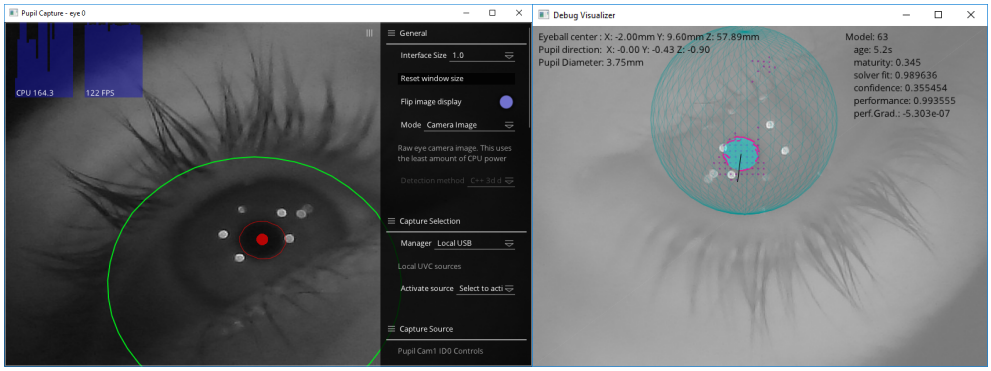

Figure 2.2: Left: Dark pupil eye image that is used to generate point of regard by an eye tracker. The red circle highlights the pupil detection, and the green circle is used to fit the shape of the captured eye. Also seen are the ring of corneal reflections produced by the IR emitters attached to the eye camera. Right: A visualization of the 3D model that is created of the eye. Note the gaze vector that extends out of the pupil, as this is used to generate the gaze point in 3D space. Images produced using Pupil Lab's capture software [16].

when fixating as there is noise and other eye movements, but overall stay centered around one point. Thus, viewing a scene is accomplished through a series of fixations that occur over relevant regions. Depending on task we produce 2-4 fixations per second [20]. Fixations over an image can be seen in the scanpath visualization in Figure 2.3.

Figure 2.3: Common visualizations of eye tracking data. Top: A scanpath shows fixations over time, with each circle indicating a fixation. The size of the circle indicates the duration of the fixation over that region, and they are separated by straight lines in which a series of saccades occurs. Bottom: A heatmap visualization represents the areas that were attended too most during the viewing sequence. Red indicates the areas that were attended to the most, and with green and then blue representing the areas looked at least. Visualization produced in SMI BeGaze analysis software.

Separating these fixations are sequences of saccades and other Rapid Eye Movements. Saccades are jerky eye movements that occur before a fixation, redirecting gaze from one location to another. Similar to fixations these eye movements can be produced overtly by forcing our eyes to move in a given direction, or involuntarily when reacting to movement or naturally viewing a scene. Saccades can can last from 30 to several hundred milliseconds, and can have a max velocity of at least 500 deg/s [20].

Table 2.1: Characteristics of Fixational and Rapid Eye Movements. Adapted from Table 1 of Lappi [20].

Eye movement type		Eye in Head Velocity Range*	Amplitude*	Duration*	Frequency*	Function
Fixational Eye Movements (FEM)	Fixation	N/A	N/A	> 100 ms,	~2-4 /sec	Stabilizes gaze on a target
	Drift	0.2-2 °/s	Few arc min, <0.13°	Depends on number of saccades, fixation duration	Same as fixation duration, besides micro-saccades	Thought to prevent neural adaption and similar to OKR/VOR stabilization
Rapid Eye Movements (REM)	Micro-saccade	~10-120 °/s peak vel.	0.1-0.3°	~10-20 ms	< 1 /s	Gaze shift between nearby targets on the same object, or close together
	Saccade	125-500 °/s peak vel.	0.1- 90°	10-300 ms	~2-4 /sec	Rapid shift in gaze between locations

* due to multiple researchers proposing different values and ranges for eye movements the most general is presented here. These values can vary based on task and environment.

Camera based eye trackers are used to study eye movements using either

Table 2.2: Characteristics of Slow Eye Movements. Adapted from Table 2 of Lappi [20].

Slow Eye Movements (SEM)	Eye in Head Velocity Range*	Function
VOR	Peak velocity of 800 °/s	Compensates for head movement to stabilize gaze, usually during fixation
OKR	Peak velocity of 30 °/s	Stabilizes gaze relative to world by compensating for retinal image slip during fixation or tracking
Pursuit	115-150 °/s	Stabilizes gaze relative to a moving target during tracking

* due to multiple researchers proposing different values and ranges for eye movements the most general is presented here. These values can vary based on task and environment.

dark pupil or bright pupil techniques. An example of this can be seen on the left in Figure 2.2. Eye trackers continuously track the movement of the pupil relative to camera, and project point forward into the world.

2.1.1 Laboratory Eye Movements

While fixations and saccades describe general eye movements in sedentary or active tasks we can further break down eye movements into a more complex hierarchy of classes. Termed laboratory eye movements these complex movements are better thought of as unique ocular events that provide a given biological function, occur under specific conditions with certain properties, and that produce particular neurological responses [20]. These eye movements

are best studied in controlled experiments where stimuli are presented without external interference, and predictable responses of the human visual system occur. These movements can be extremely short in duration and difficult to detect. Outlined below are some of these events.

When attempting to fixate on a particular point small scale Fixational Eye Movements such as micro-saccades, drift, and tremors contribute to forcing our eyes and thus fovea to remain fixated and still [20]. Specifically, micro-saccades only last between 10 and 20 milliseconds, and span less than one visual degree [27]. Drift and tremor movements are even shorter in duration than micro-saccades and detection of these events requires high frequency eye tracking data. Eye movements are ballistic in nature, and thus micro-saccades also function as a correction for when saccades overshoot a target [32]. Micro-saccades have also been shown to aid in fighting neural adaption and retinal fatigue, but results vary based on the experiment with many believing that they are a product of laboratory conditions themselves, and do not occur similarly in natural unconstrained environments. The reason for this debate is that in cases where head movement is restricted the visual system must still make corrections or jitters to ensure the that objects remain visible and centered in the fovea. In natural conditions the head contributes to this correction as the two motor systems work together [26].

The smallest of all FEMs is called tremor. The movement associated with tremors is extremely fine, matching that of the diameter of a cone in the fovea of the eye [25]. Tremors have a frequency of 90 Hz, which is slightly more

than 11 milliseconds for duration. The function of these tremors in regards to maintaining fixation is still being researched, and due to their small amplitude and quick duration are hard to identify when analyzing gaze data.

Drifts play a similar role to micro-saccades as they maintain visual fixation, but on a smaller scale then micro-saccades, and specifically in the absence of micro-saccades [25]. Drifts are slow compared to other FEMs and as with tremors can be a result of noise associated with neurons firing within the ocular system [4].

As mentioned in Section 1 Vestibulo-ocular reflex, or VOR, is another eye movement where the eyes rotate opposite of the head and effectively stabilizes the current point of regard in the world. The most common example of this is fixating on your thumb or any point in space, and trying to maintain fixation while moving your head side to side. You'll find you continue to fixate on the same point, with your eyes automatically compensating for the head movement. VOR is generally referred to as a compensatory eye movement (CEM), as VOR exists in more natural situations as well stabilizing gaze, posture, and the feeling of self movement [20]. VOR is activated by detecting motion using linear and angular accelerations detected by the inner ear.

Next there is another event known as Optokinetic nystagmus (OKN). OKN consists of first a slow phase that follows the flow of motion in the viewed scene followed by quick phase saccades which reorient gaze. This is further broken down into Stare and Look OKN [23].

Stare OKN occurs naturally when no object is being tracked, consider looking out a train window and seeing the landscape "move" away from you as you speed past. Mechanically Optokinetic reflex (OKR) is activated due to the slip (difference) in retinal image that occurs from motion. The reflex itself should not be confused with OKN, but is part of the slow phase associated with both Stare and Look OKN [20]. OKR is similar to VOR in that detects motion, except using optical motion of the retina instead of internal vestibular motion. The quick corrective saccades combat this slow adjustment of OKR to return gaze to the original position.

Look OKN can be described opposite of Stare OKN, consider you are watching a train move past you while trying to stare at a specific rail car. Stare OKN features multiple slow phases with small amplitudes with few quick phase corrective movements [18]. Since we are tracking the rail car OKR occurs following the motion of the object, and thus not many corrective saccades are needed to continue tracking the object. This is similar in function to smooth pursuits, except the response is more mechanical in nature, with smooth pursuits using higher order information.

This leads to the next classification of eye movements called smooth pursuit, generally defined as tracking. It's called tracking as the eye movement's function is to keep gaze aligned with the object is moves within the visual field. Because of this, smooth pursuits resemble fixations functionally by keeping the the point of regard fixed with the object, and saccades mechanically as they are quick and reflexive rotations of the eye [20]. Contrary

to rapid saccades which are jittery and ballistic while moving fast smooth pursuits match the velocity of the target being tracked, and follow the projected path of the object. At all stages of tracking, if the object changes trajectory a catch up saccade may occur to re-align gaze with the object and transition back into smooth pursuit. Even in the presence of occlusion humans can continue to track the object, using memory, internal representation of space and the environment, and physical properties of the object [8, 20].

2.1.2 Eye Movements in the Wild

While all of the eye movements discussed in Section 2.1.1 occur naturally, they are not easy to identify automatically, or even by hand using the characteristics established in literature. When referring to natural behavior we simply mean cases where the head and body are unrestricted, and performing a task of some sort, even if it's as simple as walking down a hallway. These behaviors range from actively navigating a 3D space to sitting and working at a desk, and include Virtual Reality environments that allow free range movement. Eye trackers measure eye in head movements, which are done in the frame of reference of the head, and we must also consider head movements and locomotion as they both play a large role in the eye movements that are performed [6, 7, 13, 19, 35]. Because of this accommodation and coordination it becomes much harder to design experiments that elicit specific movements with respect to head and eyes in the real world. Lappi (2016) presents the following seven qualitative laws for gaze behavior in the wild based on past

studies and literature [20]:

1. Gaze behavior is often highly repeatable.
2. Gaze is focused on task-relevant targets and locations.
3. Individual fixations have interpretable functional roles
4. If possible, targets are fixated "just in time", as opposed to looking at targets well ahead of time.
5. In skilled/routine behavior these just-in-time or guiding fixations are interleaved with look-ahead fixations.
6. Integration of visual and spatial information across saccades and changing vantage points does occur.
7. Gaze control is part of coordinated eye/head/body/locomotor control.

For the purpose of this paper the 6th and 7th laws are the most pertinent as they deal with the coordination between gaze and locomotion and also how spatial memory is represented across multiple coordinate systems.

2.1.3 Classification of Natural Gaze Behaviors

Commonly the above terms (fixation, saccades, micro-saccades, etc.) are used to describe physiological states of the eye and are used to describe the behavior of gaze in the real world. By doing this we end up labeling

a multitude of different states as 'fixation' and break away from definitions formed in laboratory [20]. Thus, the terminology we use to describe eye (and head) movements means something different in regards to natural behavior than it does in the experimental setting, and because of this the labeling of eye tracking data signifies a different meaning depending on the experiment itself, and the data collected.

A general flow of eye movements while performing a task can be broken up into alternating fixation-saccade sequences, where moments of fixation behavior are separated by saccadic intervals. These fixation behaviors consist of different physiological responses with functionally similar qualities. The gaze in world signal (represented by a point of regard in the 3D environment) is subject to thresholding similar to the I-DT method described in Section 2.2.1 where little to no movement in space indicates a fixation, and rapid changes indicate saccades.

Smooth pursuits and tracking appear heavily in tasks such as catching a ball, but for general navigation or free viewing tasks they do not have a significant presence. Smooth pursuits are also hard to distinguish from saccades without additional information or processing [22, 23]

Defining a fixation event is difficult as it depends on the task at hand, and in many cases can consist of different combinations of FEMs. Consider driving and looking further up the road, fixating on an obstacle, or open space. The driver may be planning the path to take, or reading a traffic sign but as you drive this point is moving both vertically and horizontally in the drivers

field of view. The fixation that is maintained on the object is actually a pursuit movement within the frame of reference of the head that matches the optical flow of the road as the car moves forward. Within the eyes frame of reference stare OKN or OKR may occur and influence the gaze in world position [20]. Due to these type of complexities it proves hard to sufficiently define fixations and saccades between environments and tasks.

2.1.4 Frames of Reference

When analyzing gaze we acknowledge that position is always recorded in relation to one frame of reference. At it's lowest level the eyes are tracked as they rotate within their socket, and return position relative to the head. In traditional remote eye tracking over a computer monitor the the eyes position relative to the screen is estimated, and gaze is then projected onto a 2D point on the screen. Thus, if we are to produce the absolute position of gaze in world we require position and orientation information about the head in relation to the world or 3D space. This paradigm of relative position and orientation follows closely that of the parent/child relationship in 3D modeling, where transformations between objects are represented through a series of matrix transforms. Within the realm of eye tracking more and more researchers have begun to solve this challenge [9, 11, 17, 38].

Beyond being a computational problem for transforming movements of gaze, we must consider that the sensory information perceived through vision and other senses is encoded in multiple frames of reference, which all are

coordinated [20]. The 3D frames of reference studied include eye coordinates, head coordinates, body coordinates, locomotor axis coordinates, and the global 3D scene coordinates.

2.2 Event Detection

One of the largest problems in using eye tracking data today comes from labeling events in this data. As mentioned in Section 2.1 events can include fixations, saccades, micro-saccades, smooth pursuit, OKN, etc. and with more rigor can be grouped into different phases and responses. As researchers design and build experiments they also build pipelines to analyze the data produced by the experiment. Most hardware vendors include analysis software tools that perform event detection (usually just fixations and saccades) and output the event labels along with raw gaze data. However, most event labelers work as a black box, restricting user control over what algorithms are used, and what is going on. These tools provide parameters for users to modify how events are detected and labeled. The problem is these parameters depend on task, eye tracker accuracy, stimulus, sampling rate and other human factors making it hard to extend established parameters to other environments.

In the eye tracking field today there are many permutations involved in gaze data itself, including sampling rate, head mounted glasses vs remote, and device specific trends such as noise and calibration accuracy. This creates a fundamental problem with generic gaze event detection and begs the question, should there be one algorithm that rules them all and works with all

forms of gaze data, or should we build specific classifiers for specific scenarios? Research in the field has been centered around building specific classifiers due to limitations in collection and labeling of data for training. Researchers generally only have one brand of eye tracker available, hand labeling of gaze data is extremely tedious, and many times there are subtle variations in the labels provided by experts [45]. Since the proposed classifiers were trained on data from a specific tracker and experiment/stimuli they end up learning subtle features present in the data and produce promising accuracy rates for their data, but would fail when processing data from a different device, subject, or even task.

2.2.1 General Algorithms

Gaze event detection is not new, and many methods have been proposed for accurately classifying fixations and saccades such as those presented by Salvucci & Goldberg in 2000 [34]. They present analysis on Dispersion-Threshold Identification (I-DT), Velocity-Threshold Identification (I-VT), Hidden-Markov-Model Identification (I-HMM), Minimum-weight-Spanning-Tree Identification (I-MST), and Area-of-Interest Identification (I-AOI) which are all common models for fixation/saccade detection.

Gaze data is presented as a representation of position over time, and is commonly provided as a 2-dimensional coordinate (x, y). Gaze position is tracked, and the positional change over time is used to detect and label specific eye movements. An alternative to using a 2D projection to represent gaze data

is a 3 dimensional gaze vector that represents where the optical axis (fovea) is pointing.

The first model that generally comes to mind is to monitor the distance spanned by the current gaze point from prior (and future) points and use a threshold to determine if points belong to a fixation. This works well for fixations as gaze points tend to have low velocity and cluster around a specific location [34]. I-DT does exactly this with a threshold for maximum dispersion, or distance from cluster that indicates gaze is moving away from the rest of the points. It also has a parameter for the minimum duration required for a window of points to be classified as a fixation. Despite having two parameters this model is quite robust, with the maximum dispersion D being determined either biologically by measuring the number of pixels spanned in 1 deg of visual angle, estimated based on task at hand (reading,free viewing, etc.), or practically by testing different values and optimizing values to fit the task [40].

I-VT applies a similar method of velocity thresholding on the point to point velocities to identify saccade points within a sequence of gaze velocities. This method produces error in the presence of noise and performs worse on data sampled at low frequencies [34].

I-HMM utilizes a probabilistic approach to classifying gaze points based a sequence of observed events and likelihood of belonging to a fixation or a saccade [33]. This model improves upon I-VT in terms of accuracy and proves to be more robust in the presence of noise [34].

I-AOI takes advantage of Areas of Interest (AOIs) in the classification of fixations. AOIs are defined by a bounding box, and fixations over a specific AOI are calculated as consecutive gaze points within the AOI. A minimum duration of time within the AOI is enforced by filtering out sequences shorter than this duration. Compared to the previous methods I-AOI performs poorly as it does not filter gaze points within the AOI for saccades, and requires AOIs to be provided [34]. While not being ideal for fixation detection AOIs have provided a valuable form of analysis through such metrics as dwell time and number of revisits [14].

2.2.2 Hand Tuned Event Detectors

As time went on more complex models for event detection were proposed, some including more advanced eye movements such as glissades, post-saccadic oscillations (PSOs), and other compensatory eye movements [23, 28, 29]. These methods generally require pre-processing and multiple phases of clustering, filtering, segmentation, and detection of extraneous events such as blinks. As more complex techniques are developed more models become specialized detectors, or only classify certain cases or events. This makes general goal of event labeling difficult if multiple detectors must be managed to create accurate global labels.

One such event detector presented by Larsson et al. is designed to identify fixations and smooth pursuits in high frequency (500Hz) eye tracking data [21]. In this model only fixations and smooth pursuits are classified with

10 different parameters. Parameters include window size and overlap, as well as measures for how consecutive points move in the same direction, and spatial thresholds.

Figure 2.4 shows another example of the NH model, featuring 7 different components and 15 parameters while only classifying saccades and PSOs [23].

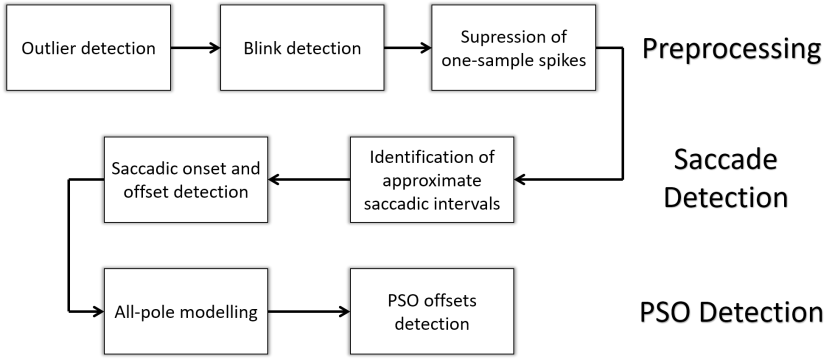

Figure 2.4: A model flow for detecting saccades and PSOs using hand tuned parameters. Note this requires 7 steps to accomplish, with 15 parameters to adjust. Adapted from Larsson 2013 [23]

.

2.2.3 Machine Learning Approaches

More recently with the rise of machine learning eye tracking researchers have explored the use of these techniques to perform event classification. Prior work has shown significant increases in accuracy compared to I-VT, I-HMM and have produced accuracy comparable to that of expert coders [1, 45].

Raimondas Zemblys analyzed 10 different machine learning algorithms including SVMs, Gradient Boosting, Random Forest (RF), and K-NN with high frequency eye tracking data and a controlled level of noise introduced to

the signal [44]. Of these 10 algorithms Random Forest performed best, also outperforming state of the art hand crafted algorithms NH and LNS [2, 23, 44].

A further analysis of RF techniques was performed by Zemblys et. al where a variety of features and feature combinations were tested alongside the effect of sampling rate, noise level and the number of trees used on classification accuracy [45]. Results from this paper showed that the median difference, mean difference, and standard deviation in gaze position were the three strongest features and only 6 features are needed to achieve optimal performance when classifying high frequency gaze data. Lower frequency gaze data has more error in classification with a decrease of approximately 0.15 in accuracy (normalized) when dropping from 120Hz to 60Hz [45].

Other methods have taken cues from signal processing and generate machine learning features based on shape analysis of gaze amplitude. Vidal et al. produced a classification rate of 92% for smooth pursuits in a lab experiment that required tracking of an on screen circle moving at predetermined speeds and distances [39]. A trend seen in these types of models is that instead of building overly complicated detectors the bulk of the work is to produce meaningful gaze features that are then selected using feature extraction techniques [39, 45]. Features used represent gaze behavior temporally and spatially with variations in results at different sample rates and noise levels.

Another approach is to apply neural networks and other deep learning techniques as traditional methods and features become exhausted. For example, a method presented by Anantrasirichai et al. utilizes convolution

neural networks to extract features (activation kernels at each layer of the CNN) from image regions around a set of selected fixation points, and trains an SVM using these and other gaze based features to perform fixation and saccade detection [1]. The concept here is that the underlying image content has an effect on whether eye movements near this point tend to be fixational or saccadic. This is an encouraging approach as it is well established that task at hand plays a role in dictating where our eyes visit and how our scanpath is constructed [37]. Results with this model were promising achieving $> 80\%$ accuracy in all models that employed these local features, and 86.5% accuracy with the full model. I-VT and I-HMM produced an accuracy of $\approx 70\%$ given the same data.

So far most of these methods require pre-processing, complex features, or other work that must be done to the data before it can be classified. Hoppe et al. has proposed an end-to-end model, one which directly takes gaze data and directly feeds a single network producing a usable classification. This model uses frequency domain analysis on the gaze signal to feed the network [15]. The output of the CNN is a set of probabilities indicating the chance of being a fixation, saccade, or smooth pursuit (the maximum probability indicating the classification). Some may argue this solution is not perfectly end-to-end, i.e window size for the FFT must be defined, and the FFT spectrum must be calculated around each gaze sample. The model outperformed I-DT and I-VT in specialized classification (i.e. is it a fixation or not) and had some nice results in the multi-class setting featuring an F-score

of 0.7 for fixations, and an average F score of 0.55 over all classes [15].

Evident from these models is the fact that there still isn't a one size fits all classifier, and researchers are still proposing new features and models for the most robust event detector. One bottleneck with using supervised machine learning algorithms for this task is the fact that ground truth data is sparse, usually associated with a specific task or function, and is overall very time consuming to produce. As more data is collected and aggregated there is hope that deep learning techniques can be utilized to build more complex networks (the above examples used only three or less convolution and pooling layers) and build a truly robust classifier. Similar techniques are now being employed in gaze estimation using large sets of synthetically generated gaze data [42]. In this case images of the eye are rendered and labeled using the generated gaze vectors. By training a model on this large corpus of data and matching input real eye images to the closest synthetic image has produced accuracy within 10 deg, which is impressive for calibration and eye model free estimation.

2.3 Inter-rater reliability

For the labeling of eye movements in gaze data experts typically have years of experience with eye tracking data and research in human vision. In general we expect these experts to agree on a large percentage of labels, with intermittent errors [2]. However, event detection algorithms do not provide this high level of agreement [45]. A visualization of this can be seen in Figure 2.5 where labels from 4 experts are compared side by side. These experts agree

on the majority of the labels, with two experts marking saccades that that the other two didn't.

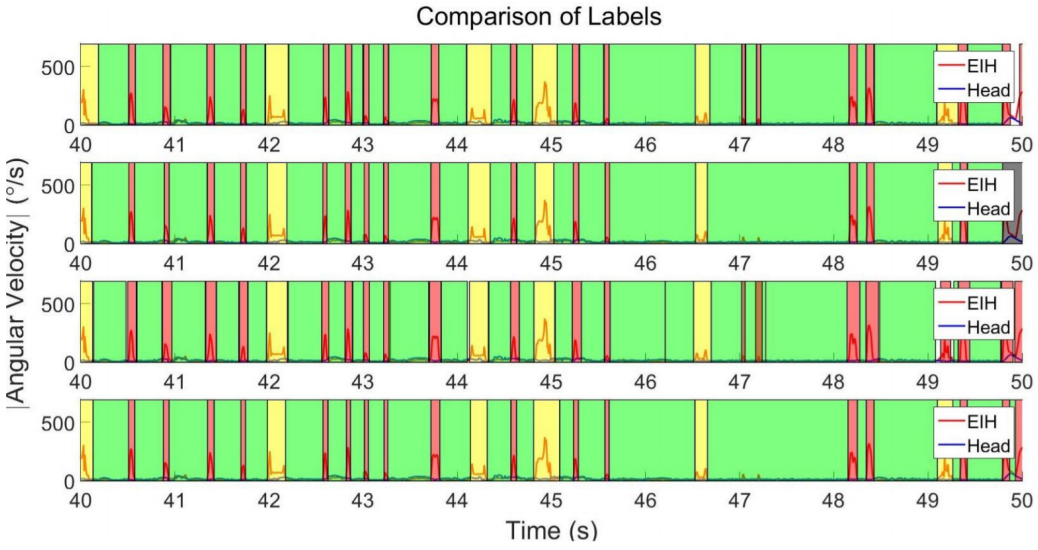

Figure 2.5: Event labels for 10 seconds of gaze data from 4 different experts, green represents fixation, yellow indicates blinks, and red indicates a saccade. Notice that experts agree on most events, with slight variations in the starting and ending points.

While accuracy, precision and recall are common metrics for assessing the accuracy and reliability they operate on a sample-to-sample comparison. With data such a gaze events many expert coders will ultimately agree on the presence of an event, but show slight differences in duration or start/stop boundaries for said event. This causes a high calculated accuracy to convey that two experts labels agree, but these subtle differences add up and have equal weight to the correct cases.

Instead, when measuring the agreeance of two raters we can use Cohen's

Kappa to assign a value between 0 and 1 where 1 represents perfect agreement, and 0 represents exclusive disagreement [5]. Cohen's Kappa is defined below, where p_o is the proportion of samples of which the two raters agreed over the total count of samples, and p_c represents the probability of chance agreement between the raters based on the how the labels are distributed among classes for each rater.

$$\kappa = 1 - \frac{1 - p_o}{1 - p_c} \tag{2.1}$$

The calculated κ is then used to measure the agreement between expert coders, as well as to measure the success of an automated event detector compared to an expert coder [45].

When extending this idea to more than 2 raters we can use a model that accounts for each different rater, such as Fleiss' Kappa or other multi-rater algorithms [12, 31].

Chapter 3

System Design

Our goal is to collect real world data from subjects, and then using information about their eye movements collect a naturalistic dataset in VR. This section outlines the real world dataset used to drive the experiment, the underlying methodology, and the implementation.

3.1 Real World

Eye tracking data was collected from 3 subjects performing 3 different real world tasks, walking indoors, walking outdoors, and playing catch. For the indoor walking task subjects walked through multiple hallways, and went up and down stairs. For outdoor data collection a wooded nature trail was used. There was no instruction for each of these tasks beside walking. When playing catch a tennis ball was thrown between the subject and several others in a lab environment. 5 minutes of eye tracking data was collected for each task.

Manual event labeling was performed on 1 subjects data by 8 different "experts". These experts labeled the majority of the walking task, with some labeling more data than others. The GUI can be seen in Figure 3.1, which

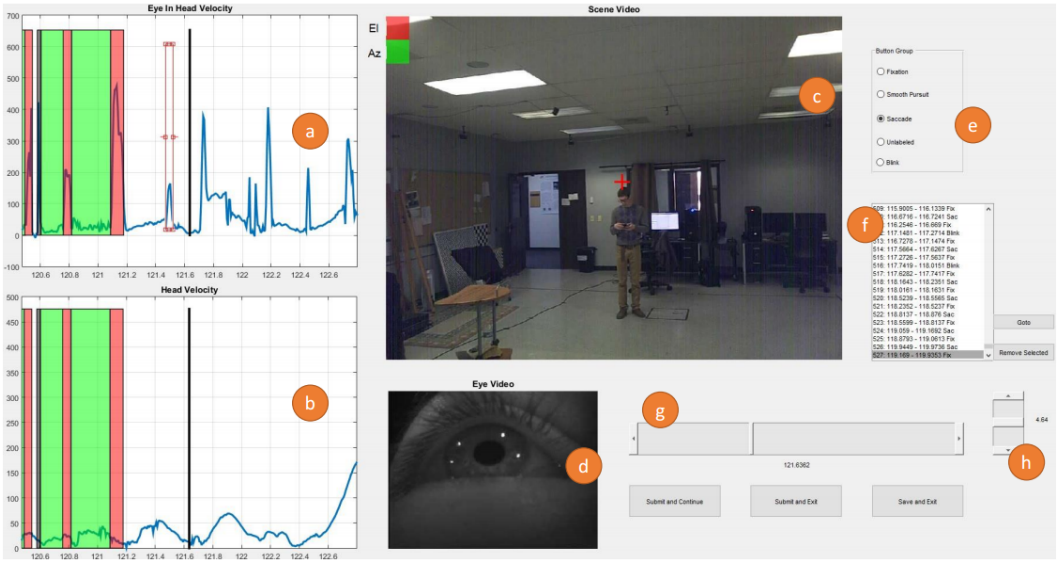

Figure 3.1: The Matlab labeling GUI used by experts to classify saccades, fixations, blinks, and smooth pursuits. (a) Eye in Head velocity signal vs time, (b) Head velocity signal vs time, (c) Scene video from eye tracker with current gaze point overlaid (d) Eye image for corresponding video frame (e) List of event types for labeling, (f) list of user entered labels, (g) temporal slider for navigation through the data, and (h) a slider for time scale to allow bigger or smaller jumps through the data.

was used by the experts to mark fixational eye movements, saccades, smooth tracking movements, and blinks. By labeling blinks we can easily remove them from data. We can easily export all of the fixations and saccades as well, since the start and end times are saved during the labeling process. The data extracted for input to the VR game is a description of saccades and fixations that occurred, based on the manually labeled data.

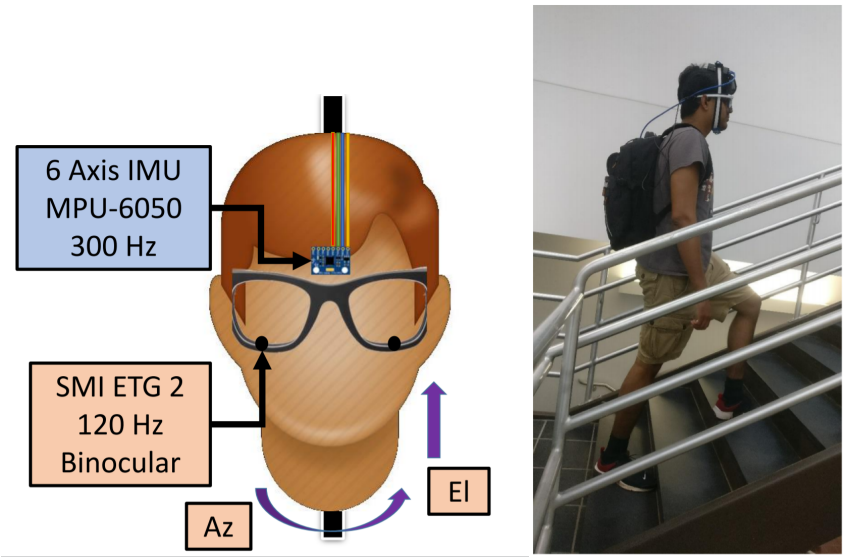

Figure 3.2: Left: Equipment setup for the real world experiment. Subjects wore a pair of SMI 120 Hz Eye Tracking Glasses 2 along with a InvenSense 6 Axis 300 Hz IMU that tracks head orientation. Right: Subject going up stairs during the walking task.

3.2 Study Overview

For data collection a 3D VR environment is created and presented using a head mounted display. Motion tracking is used to record the user's position and orientation as they move through space, as well as tracking a game peripheral that is used to interact within VR. Gaze data collected from the described natural task is used to produce the stimulus presented in VR, and is read from an input file at the start of the experiment. Specifically we extract fixation durations, and the elevation and azimuthal visual angles (vertical and horizontal respectively) covered by saccades, along with duration and eye in head velocity. We can then sample the points in this distribution

and animate a target with the given motion to produce a dataset of 'natural' eye movements.

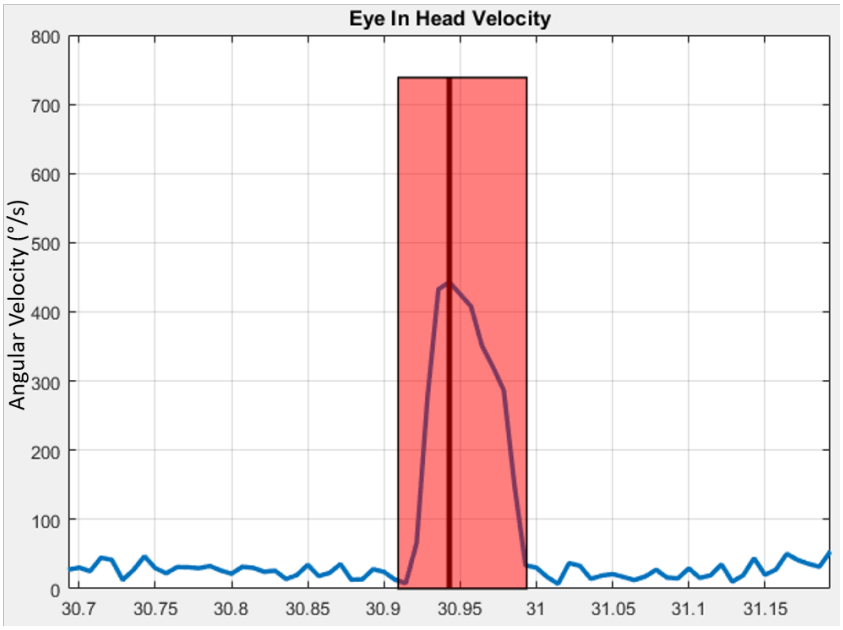

Figure 3.3: An example of a saccade in eye tracking data, indicated by the red labeled region. The graph represents the instantaneous angular velocity of gaze over time. The saccade is easily identified with the peak in velocity.

For saccades an example can be seen in Figure 3.3. The velocity curve is one of the inputs to the system so that saccades can accurately be recreated.

Eye tracking data is recorded while the subject performs a series of trials, with each trial representing a specific saccade or fixation. The user is presented with a game like interface where points are scored for successfully targeting the sphere with a hand-held controller, as seen in Figure 3.5.

The player is incentivized to track the ball with their eyes in VR as

Figure 3.4: Image of HTC Vive VR HMD and controller used for motion tracking for the experiment. IR cameras are placed on the ceiling surrounding the capture space of the experiment. Not pictured is the Pupil Labs eye tracker [16] installed within the HMD.

they are told to shoot the ball for points. When the target is green the player earns 10 points by aiming the gun and pulling the trigger, however when the target is red shooting it results in the loss of 5 points. There is no penalty for a missed shot.

As useful as the raw data produced by this experiment is we can also see if the data collected during our trials match the saccades that were desired, and could be extended top produce blocks of gaze data that are already labeled. This is useful for prototyping machine learning algorithms without the need to spend time labeling data.

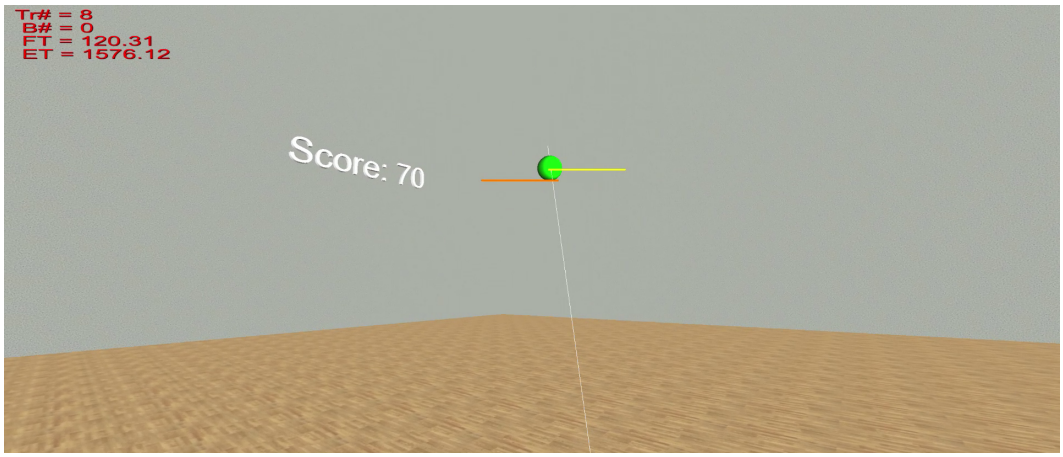

Figure 3.5: Screenshot of the virtual reality environment. The controller is used to aim and shoot at the targets using the trigger located on the controller grip. Note: The screen shot is not representative of the field of view in the HMD and the gaze vectors are not visible in the HMD.

3.3 Gaze Distribution Processing

The data collected from the real world experiment (described in Section 3.1) generates raw gaze data. For the purpose of this work we analyze the eye movements in terms of a 3D gaze vector changing over time. Before the gaze velocity data is presented for labeling in the Matlab GUI the gaze data is median filtered then wavelet de-noised up to 4 levels [10].

Next, the recorded classes from all labelers all of the saccades and fixations are exported to their own files. From this one file describing fixations and one describing saccades is produced for input to the VR game.

First, fixations are easy as the file simply contains duration values, one fixation on each line. The saccades require more information, but also include duration. With saccades we present the target in the middle of player's field

of view, and move the target in the same direction as the saccade did. This direction is determined based on the angular change that occurs in the gaze vector from the first point in the saccade to the last. The horizontal and vertical angular change is used to determine the ending point for the target.

To animate the object we have all of the tools needed, a starting point (in global coordinates), an ending point in global coordinates, and a duration. The easy way to animate this is linearly interpolate position from start to finish over the duration. This however does not preserve the velocity profile of the intended saccade. One way to fix this to define an alternate interpolating function, one that follows a linear path, but reflects how quickly the eyes were moving during the saccade. Interpolating functions are defined as $interp(a, b, t)$ where a is a starting position, b is the final position, and t is a normalized percentage of time that has elapsed. $t = 0$ represents the beginning of the animation and $interp(a, b, 0)$ would return a. Likewise, $t = 1$ represents the end of the animation and would return b.

To define this interpolating function we must design a function F that takes t as input and transforms this to a value represent how much distance had been covered by the eye movement at that point in time, essentially displacement from the starting point. For example, if 25% of the distance is covered in the first half of the saccade then $F(0.5) = .25$. The output of F is then fed to the traditional linear interpolation formula, $interp(t) = a * (1 - t) + b(t)$, to determine the current position. Thus each line in the saccade data file contains duration, vertical angle (ϕ), horizontal angle (θ), and the four coefficients that

define the interpolating function F.

To generate the cubic coefficients we first define $F(t) = \frac{\int_0^t v(t)dt}{D}$ where v(t) is the instantaneous velocity at time t and D represents total displacement, $D = \int_0^1 v(t)dt$. Since $F \to [0, 1]$ and represents the amount of distance covered at time t it can be used as the interpolating function. F is computed as a discrete function using numerical integration and the same sample rate as the velocity data. Instead of saving the whole function to file we instead compute an interpolating cubic function $F_{cubic}(t) = a * t^3 + b * t^2 + c * t + d$ that fits that data of F and save the 4 function coefficients. An example F and F_c can be seen in Figure 3.6. Using this method aggregated over all of the labeled saccades produced a root mean squared error value of 0.0165 in normalized units, meaning the estimating functions only introduced 1.65% of error over the whole dataset.

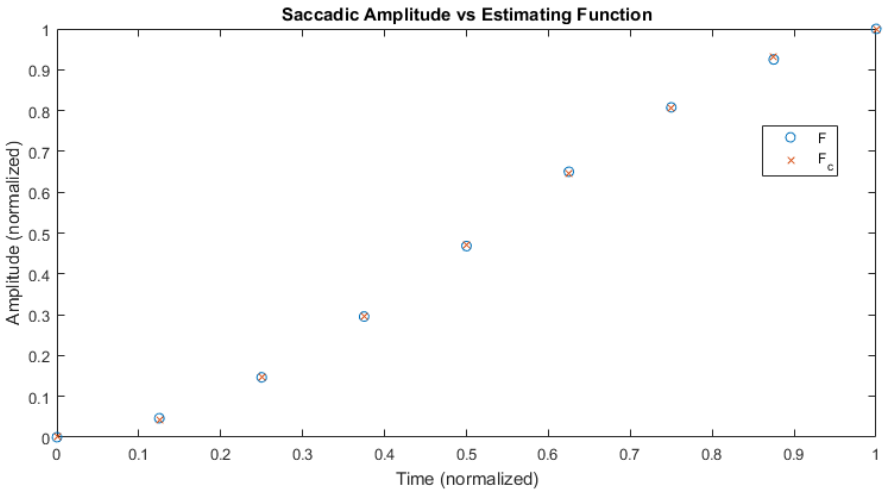

Figure 3.6: Plot of F and F_c functions for one saccade.

3.4 Hardware & Data Collection

The Windows PC used for the experiment features a Intel® Core™ i7-6700 quad core CPU running at 3.4Ghz with an Nvidia® GeForce™ 980 GPU. VR and motion capture are accomplished using the HTC® Vive™ HMD and controller. The headset can be seen in Figure 3.4.

The software used to render the VR environment and control experiment flow is controlled by version 5.7 of the Vizard virtual reality toolkit from WorldViz. Vizard executes python scripts, which is the where all of the code for the experiment lives. Matlab was also used for processing, analysis, and resulting visualizations.

To collect eye tracking data the Pupil Labs HMD binocular eye tracker was used. The eye tracker streams gaze data for both eyes at 120Hz, but data logging was limited to the HMD refresh rate of 90Hz.

For this experiment there are two phases, the first featuring longer pauses before and after the animation of the target, and a second phase with much shorter pauses. The short phase featured 1.5 seconds of delay before animation, and 1 second delay after. The fast phase had a 0.3 second delay before animation, with 0.2 seconds after animation.

3.5 Eye Tracking Calibration & Validation

The eye tracking calibration was performed before the experiment began using the procedure provided by Pupil Labs in Unity. 14 targets were

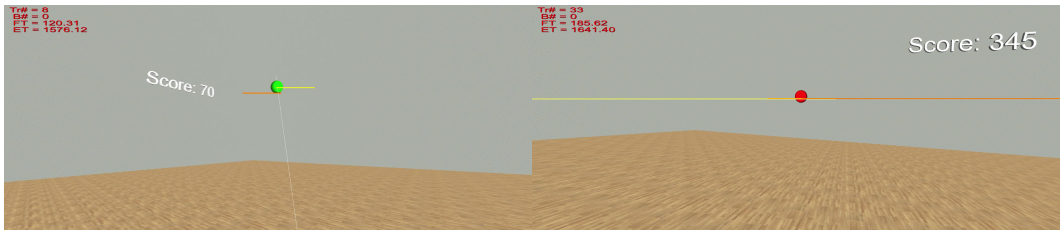

Figure 3.7: In game screen shot showing both the green and red target cases. Shooting the green sphere earns the user 10 points, while shooting the red sphere causes the player to lose 5 points. Note: The screen shot is not representative of the field of view in the HMD and the gaze vectors are not visible in the HMD.

presented to the user at one depth value. This can be seen in Figure 3.8. 120 data samples are collected from the eye tracker for each target, 30 of these points are removed as outliers and the remaining 90 are averaged to determine what position is used for the calibration. A rough validation was performed by asking the subject to look at a virtual point in their environment and ensuring the generated gaze point lines up with the target.

Through this process S9's calibration was determined to be invalid, and data was removed from analysis. This is mostly due to errors in pupil detection on the eye tracker images. For each subject several minutes were spent tweaking camera parameters to ensure that the pupil was tracked consistently and accurately. Due to the color of the iris, shape, and other characteristics of the eye some users pupils are not tracked well.

To test the accuracy of this calibration a 3D grid of points is presented to the user one at a time in Vizard, while angular error between the eye tracking data and ground truth data is recorded. Figure 3.9 visualizes this grid, and

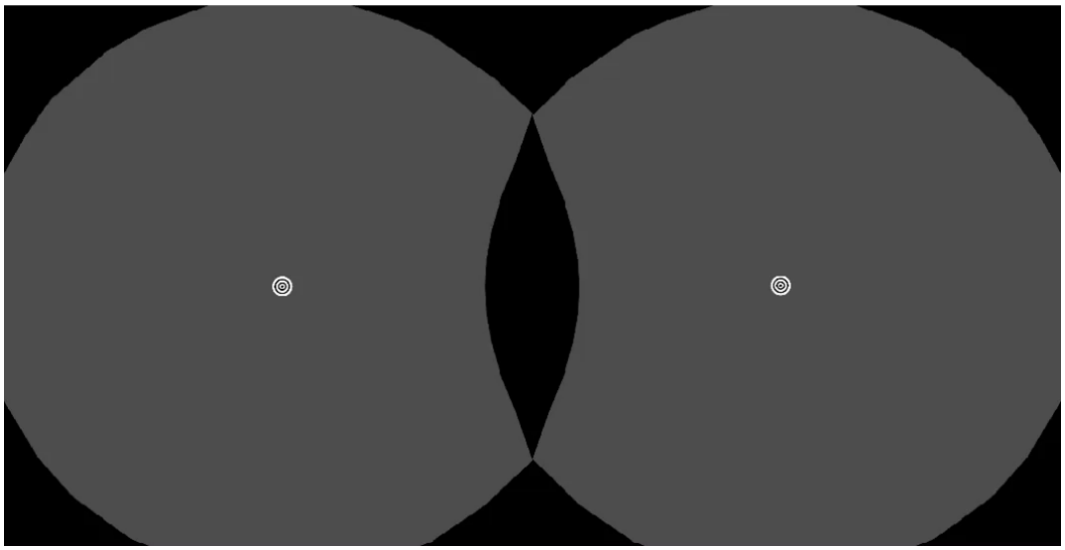

Figure 3.8: Screenshot of what the calibration process looks like within the VR headset. 14 white targets are presented at various points in the visual field. All targets were located one depth value.

shows how each point is presented to the user. The validation itself features points located at 3m, 4m, and 5m away from the subject with 9 points on each plane. Overall these points span 3.43° horizontally and 6.64° vertically in both directions. This process was only performed on the last 5 subjects. 4 of these subjects returned to perform another calibration and validation to get an idea of the consistency between calibrations.

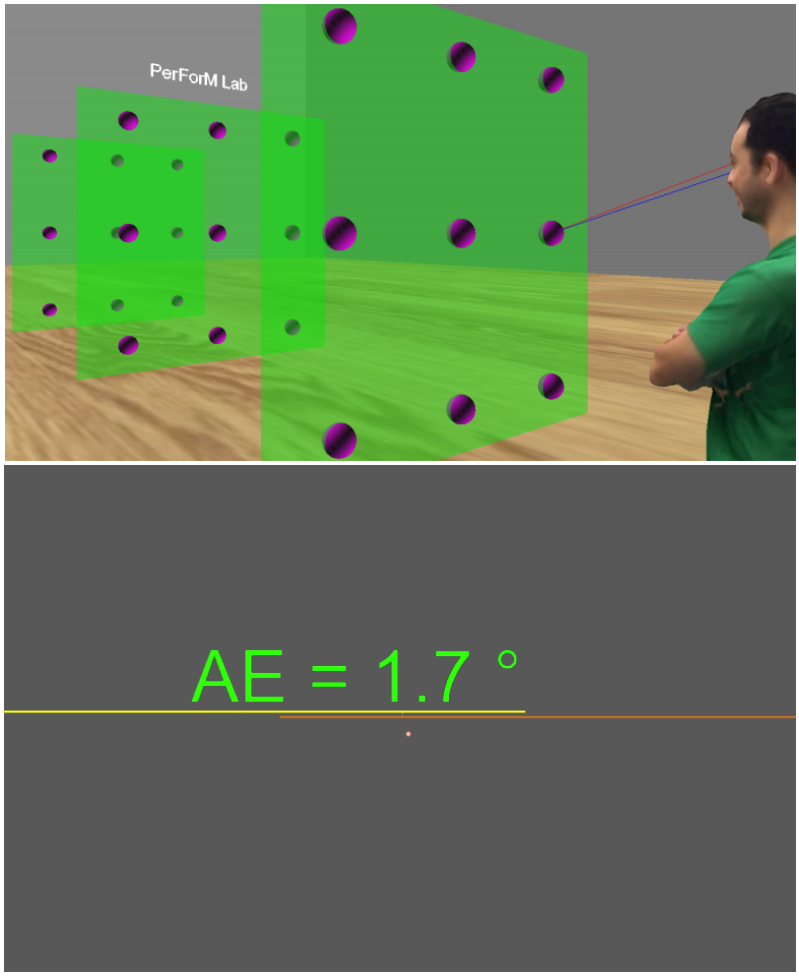

Figure 3.9: Top: Visualization of the 3D grid of points used to measure error before performing the experiment. The grid points span 3.43° horizontally and 6.64° vertically in both directions. Bottom: Screenshot from the eye tracking validation process. The yellow line represents the direction of the left eye normal, and the orange line represents the right eye normal. The average of these two vectors generates a cyclopean gaze vector which is then compared to a ground truth vector drawn to the target. Angular error is computed between these two vectors and displayed on the experimenters display. Note: The screen shot is not representative of the field of view in the HMD and only the gray background and validation point are visible in the HMD.

3.6 System Integration

Figure 3.10 shows a general overview of project. Collected data is first median filtered, de-noised, then was labeled by experts. The processed data is fed to the VR experiment as a text file, and the experiment then randomly selects saccades to present to the user while recording eye tracking, motion capture, and trial data. Through this process gaze directions are interpreted as spherical coordinates relative to the tracked head position. When modeling the targets in the 3D these coordinates are transformed into cartesian coordinates and place into the global coordinate space.

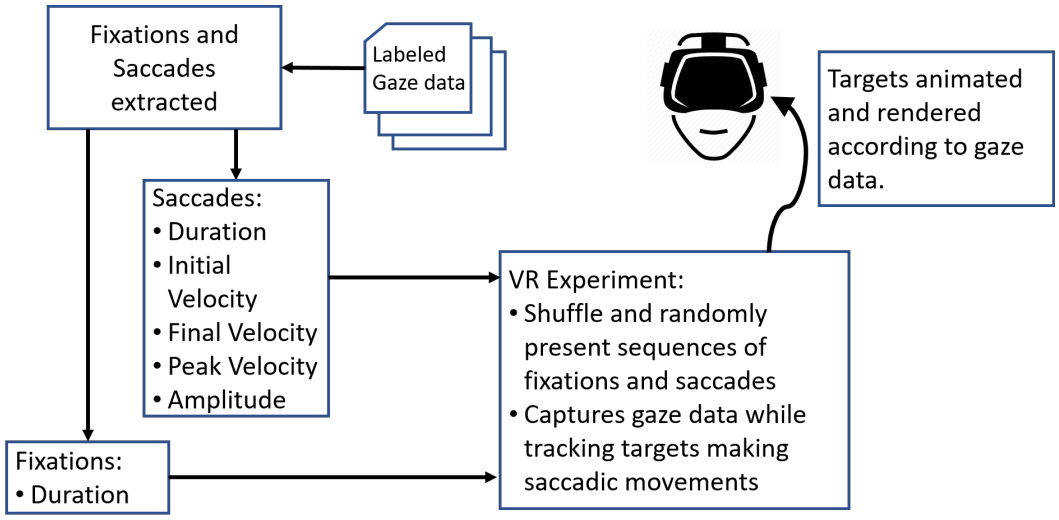

Figure 3.10: Flow diagram of gaze data as it transitions from labeled data to user's VR headset. First, fixation and saccade sequences are extracted from labeled gaze data. Next, these are fed to the experiment where they are randomized and presented using a spherical target.

Chapter 4

Results and Discussion

4.1 Subject Statistics

Data was collected from 14 subjects (10 Male,4 Female). Average age was 23 and ranged from 18-32. One subject's data was not included in analysis due to a low quality track of the pupil during calibration and before the experiment. Eye tracking error analysis is presented in Section 4.2.

Due to changes within the experiment structure not all subjects participated in the eye tracking validation, or the slow and fast phases. Out of all of the subjects 10 of them completed both the fast and slow phases, completing 150 trials of the slow case, and 200 trials of the fast case. The final 5 subjects performed an eye tracking validation, with 4 of them returning and completing a calibration/validation again.

4.2 Eye Tracking Error Analysis

For remainder of the results section eye tracking error is defined as absolute angular error calculated as the angle between two vectors. This value does not take into vertical or horizontal components and has equal weighting

in any direction around the ground truth point. The angle is calculated as

$$\theta = arccos(v_1 \bullet v_2)$$

where v_1 and v_2 are normalized vectors.

Figure 4.2 displays a heatmap of the error across all subjects at each depth featured in the eye tracking validation. The general trend of eye tracking data is that points directly in the middle of the visual field are more accurate than points towards closer to the horizontal and vertical extremes. These averages include both validations for the subjects who returned a second time. Figure 4.1 charts the error across all subjects for each of the 27 validation points.

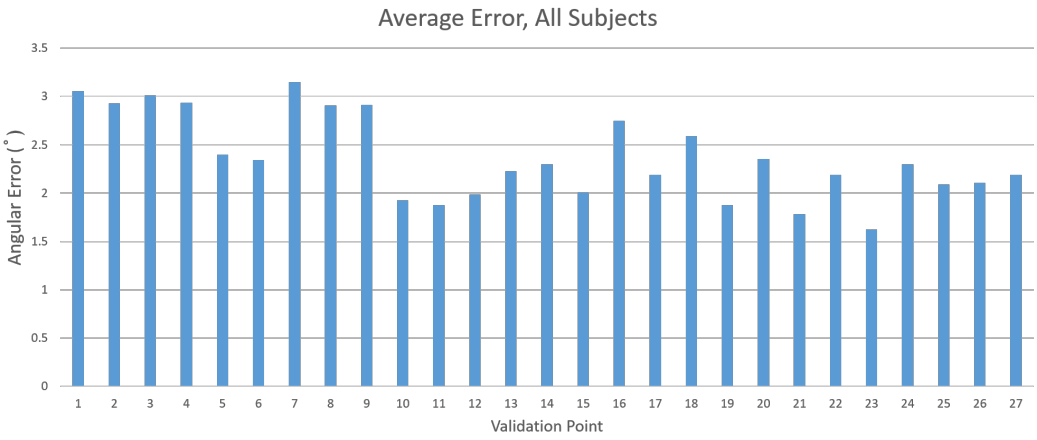

Figure 4.1: Average angular error across each subject for each of the 27 validation points.

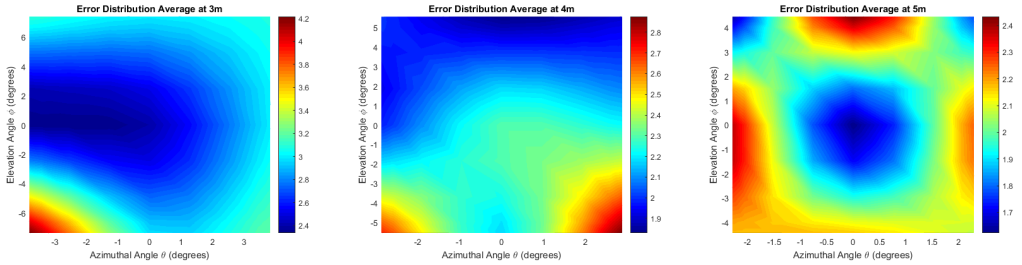

Figure 4.2: Plot of average gaze error over the visual field at 3m, 4m, and 5m. Elevation and Azimuth angles are relative to user's orientation, with values of 0,0 indicating the the target is in the center of the visual field.

Table 4.1 compares error between subjects at each depth. It is clear that highest amount of error exists in the closest set of points, located at 3 meters away. This makes sense, as the points closest to the subject span the largest field of view and we continue to see the trend of error increasing as move toward the periphery.

An inconsistency can be noticed between calibrations of each subject, and even between calibrations. An example of this can be seen in Figure 4.3 where the error distribution from each of S14's calibrations are compared. In this case specifically error was distributed similarly at each depth, which we can see in the table. The averages at each depth are nearly the same, but do trend downwards for farther depths. As you can see in this comparison the distribution of where the lowest error (blue) and highest error (red) are at each depth remains quite consistent, which represents a stable track. However, these extremes within the vertical and horizontal field vary between the two calibrations. In the top calibration error remains lowest in the right-middle of

the field of view, and in the lower plots error is lowest in the top-left.

Compare this to the error plots in Figure 4.4 of S10's data. For this subject the point of lowest error shifts depending on depth, indicating a less stable track across the visual field. For the calibration shown on the lower row the results shift wildly as depth jumps from 4 meters to 5 meters. Additionally, looking at the table you can see that in S10's first calibration the error decreased from around 3° to 1.4° at 3 and 5 meters. In the second calibration the error actually increases at 5 meters, and the values stay close to 3° with not drop in error. Note for all of these plots the heatmap is created using only 9 points, so interpolations can be misleading as to the error trend in between points, and only 4 points are positioned at angles without 0 as a horizontal or vertical component. In the future, more than 9 points can be used for each plane to better model the space between the extreme corners and the horizontal/vertical axes.

Thus, the calibrations we recorded vary from person to person, but overall are not very stable between sessions. This could in part from the calibration routine itself, or the variability in the pupil tracking algorithm. It is import to note that these error distributions may differ from that of the glasses based Pupil Labs eye tracker due to design. While the eye cameras on the eye tracking glass can be moved and rotated relative to the eye, due to being attached a fixed point on the VR headset's lenses the camera will always face the eye from a vantage point below and slightingly offset from the eye. The only variation in this view occurs when the lenses within the headset are adjust

to be closer/farther away from the eyes. This fixed camera position keeps the eye roughly in the middle of the image, but is then biased towards creating a very stable track looking straight forward or in the horizontal direction. Movement in the vertical direction brings the pupil into a smaller region within the image and complicates the pupil detection/tracking algorithm. This is one explanation for an error distribution that increase quickly higher or lower in the visual field, and also cases like the second row in Figure 4.3 where the top of the visual field is much more accurate than the lower portion. This bias comes from the specific placement of the headset for that calibration and the resulting viewpoint of eye tracking image.

Table 4.1: Average Angular Error

Subject	Avg Error: 3 M (°)	Avg Error: 4 M (°)	Avg Error: 5 M (°)
S10	3.224787	1.307519	1.401048
S10_2	3.176066	3.052775	3.754438
S11	3.158966	2.492439	1.511375
S11_2	4.251537	3.536662	3.378656
S12	3.062331	1.884004	1.846797
S13	2.437057	2.042405	1.546021
S13_2	1.836783	1.51501	1.31385
S14	1.96598	1.449646	1.432358
S14_2	2.518949	2.268936	2.331284

Average angular error recorded during the validation process. 4 subjects repeated the calibration and validation process, this is indicated with a '_2' added to the subject id in the first column.

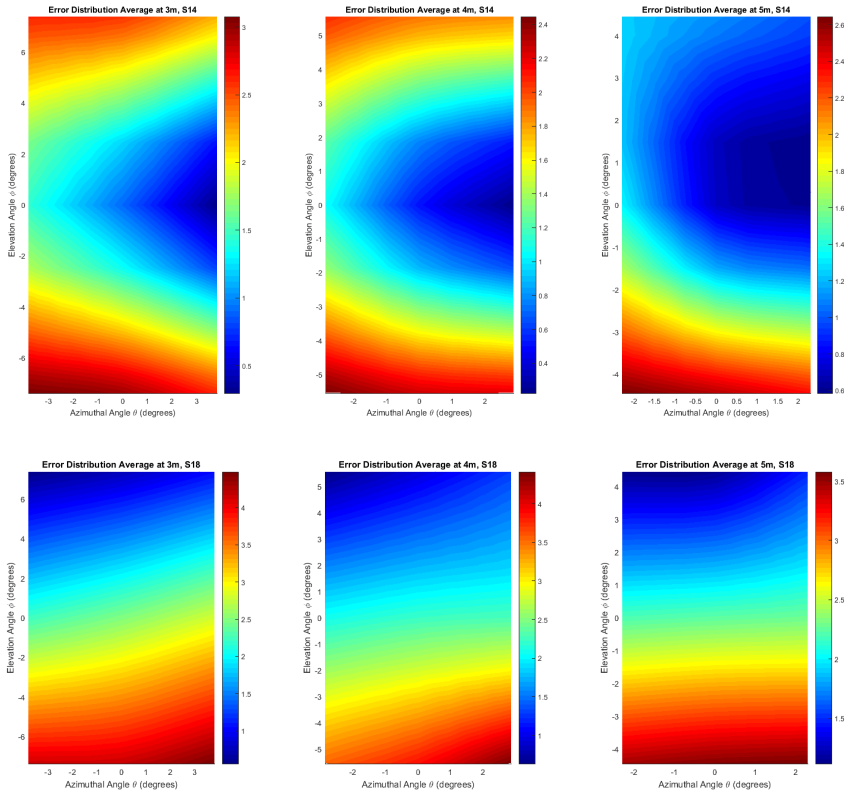

Figure 4.3: Comparison of error between two different calibrations of the same subject. Error is plotted as a heatmap across the visual field. In this case the subject's calibration remains consistent at different depths, but areas of low error vary between two separate calibrations.

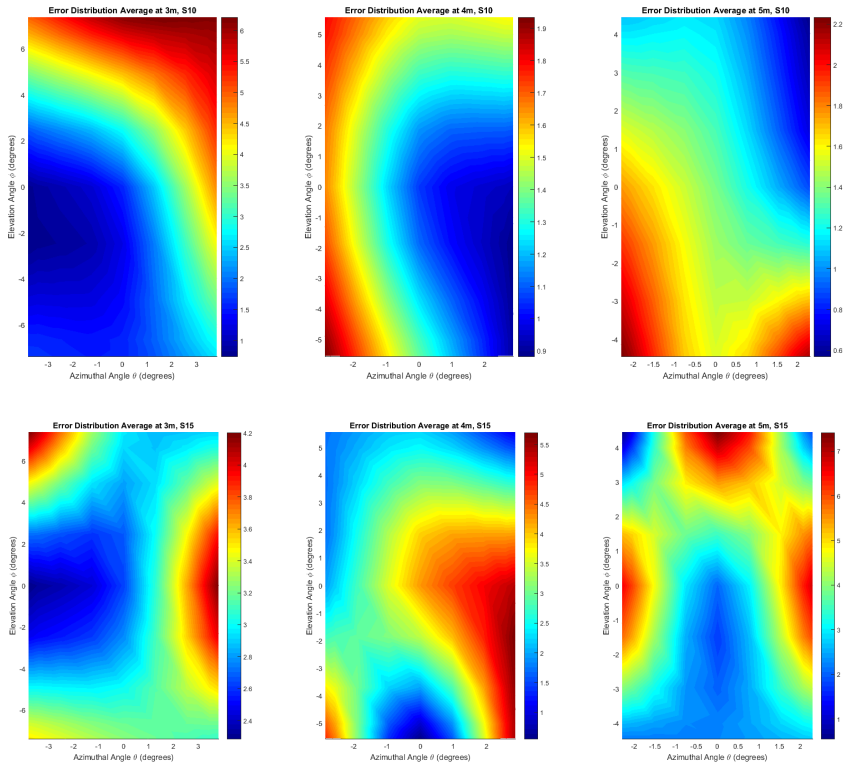

Figure 4.4: Comparison of error between two different calibrations of the same subject. Error is plotted as a heatmap across the visual field. In this case the subject's calibration is inconsistent at different depths, and between two separate calibrations.

In general we cannot say there were any clear trends in the error across all subjects, with a slight bias of a more accurate track with targets at farther distances. This can be further seen in the scatter plots of Figures 4.5 and 4.6 as points from each subject are distributed around the visual field.

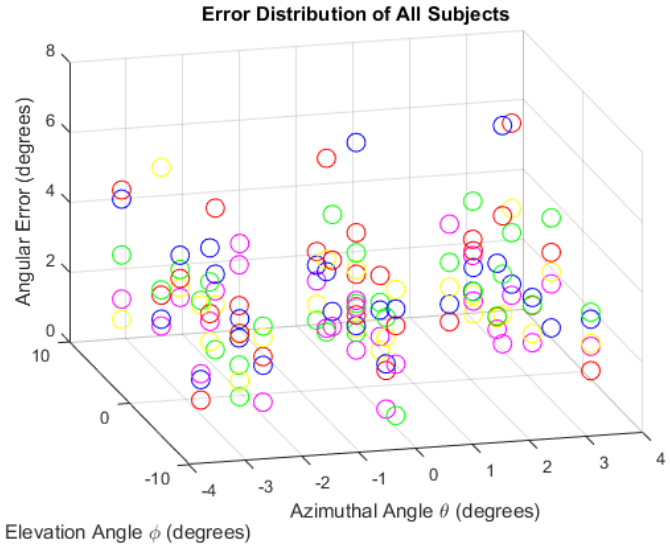

Figure 4.5: Plot of mean error values for all 27 validation points from each of the 5 subjects. Each color represents a different subject. Height indicates the angular error the trimmed mean value out of the 120 samples recorded.

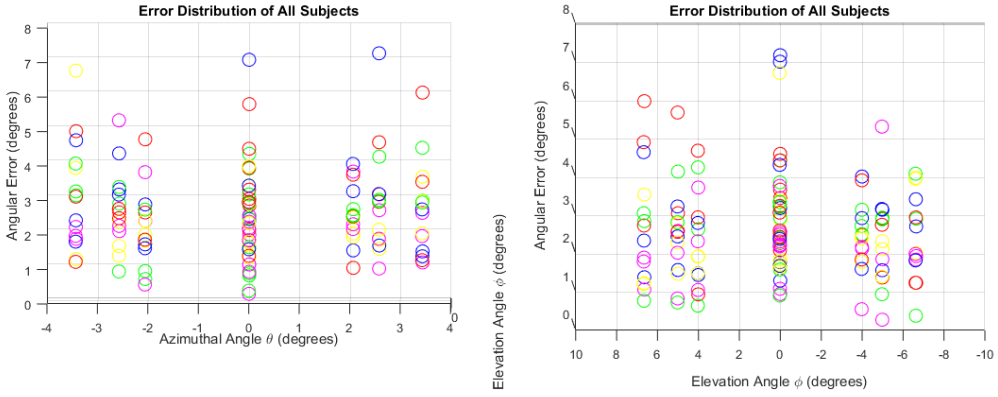

Figure 4.6: Left: Plot of mean error values over horizontal field of view for all 27 validation points, from each of the 5 subjects. Right: Plot of mean error values over vertical field of view for all 27 validation points, from each of the 5 subjects.

4.3 Ball/Gaze Intersection

If we aim to automatically identify values like reaction time, a recreated velocity profile, and the ratio of eye to head movement we need some sort of metric that indicates if the task was actually performed during the trial. This metric can be thresholded to filter out trials containing data we do not want to include in analysis. For this experiment the percentage of time the user looks at the target is used. This metric is computed by testing the intersection of the subject's gaze vector indicating where they are looking with the target itself. Finding the intersection of a ray with a sphere in 3D is easy to compute, and a tracking percentage is computed for each trial. If the subject was perfectly fixated on the sphere from moment it is animated to the moment it disappears than this value will be 100% and if they never look at the target this value will be 0%.

For a task with such fast animations we cannot expect our subjects to achieve a perfect percentage, but this value allows to determine if the trial was valid or not. One large factor in the computation of this percentage is the error present in the eye tracking signal. The process of computing ray sphere intersections does not incorporate any width or size parameter for the line, which means the computed percentage is only completely true if the eye tracker has 0° of error. This is obviously not true, as any amount of angular error would cause the intersection test to fail. This problem is exasperated by cases where humans may fixate towards the edge, or even slightly off the target they are tracking. To address this the intersection of rays that are uniformly

distributed along a cone defined by an angle around a gaze vector are tested, Figure 4.7. If the absolute angular error used to generate this cone is equal to the true eye tracking error present at the current data sample, one of the bundled rays will intersect with the target and the intersection percentage for this trial would be correct. For this dataset 32 different rays were included in each bundle.

The intersection percentage averaged across all trials was computed for a constant error angle with equally spaced values between 0 and 6 °, Figure 4.9. With the exception of S12, the 4 subjects intersection percentage trend similarly with a difference near 5% for two of these subjects. Note that these intersection percentages also rely on the size of the target within the subject's field of view, as the radius was held constant but closer targets feature more area and may provide more false positives in this computation.

This method clearly works, as the percentages at 0° of error are all less than 10%, and increasing to values of at least 25%. Somewhere along this range of error values is an optimal angle that minimizes false negatives and false positives. Because one angle is used for all of the trials a perfectly accurate percentage cannot be computed unless the correct error angle is used for each data sample within each trial. Instead a range of error angles was tested with the hypothesis that the optimal angle would be visually noticeable with a large step up in plot of intersection percentage. The trends of Figure 4.9 are actually much more smooth, but seem to flatten out with values greater than 4°. The angle step size was 0.2° which also makes identifying this step

harder. Without manually labeling the data there is no way to easily compute false positives and negatives, and the optimal value would vary from subject to subject. For further analysis a value of 3° was used for all subjects which was derived from the curves in Figure 4.9 as well as observing a maximum average error near 3° in Figure 4.1.

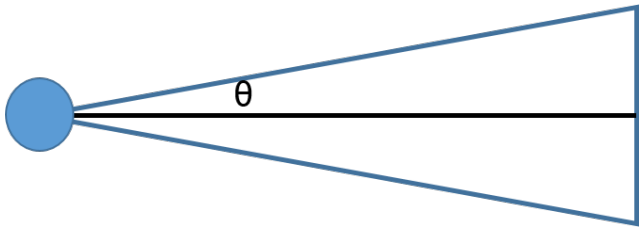

Figure 4.7: Illustration of a cone generated based on an input angle, θ. When computing gaze and ball intersection rays are uniformly distributed along this cone.

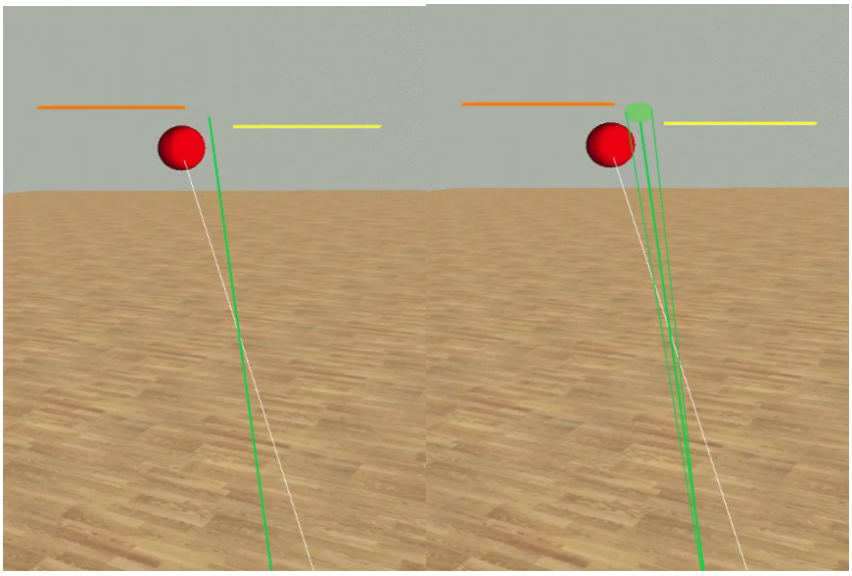

Figure 4.8: Screenshot of experiment, the white line is the direction of the controller, and the green line represents the cyclopean gaze vector. Pictured on the left is the gaze vector with no error correction. On the right the cone used to test gaze/ball intersection is drawn.

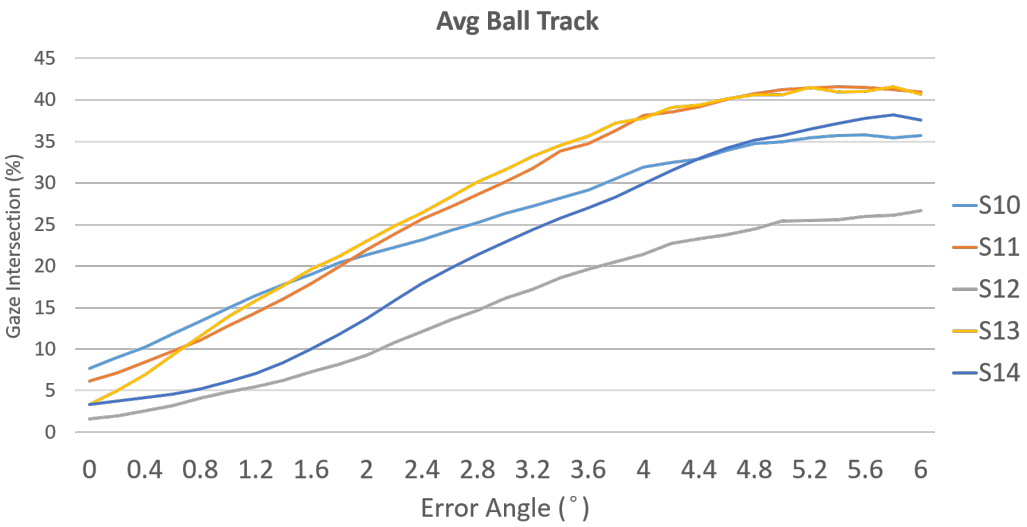

Figure 4.9: Average intersection percentage of gaze with ball over a range of error angles. Each line represents a different subject.

Using 3° of error the average gaze/ball intersection percentage during slow and fast phases was computed, Figure 4.10. Across these 5 subjects it is clear the it is easier to track the ball in the slow phase, and with the exception of S12 all produce an intersection percentage near 20%, or looking at the ball a fifth of the time over a trial. Note that S12 also had an inconsistent as seen in Section 4.2, and this effect may carry over into the intersection percentage. These results shows that most subjects were limited to looking at the fast targets they all performed differently during the slow phase. The intersection percentages range from 21% to 43% in the slow phase.

While the goal of computing gaze/ball intersection was to filter out valid trials the following analysis does not apply any filtering to the trials

based on this value.

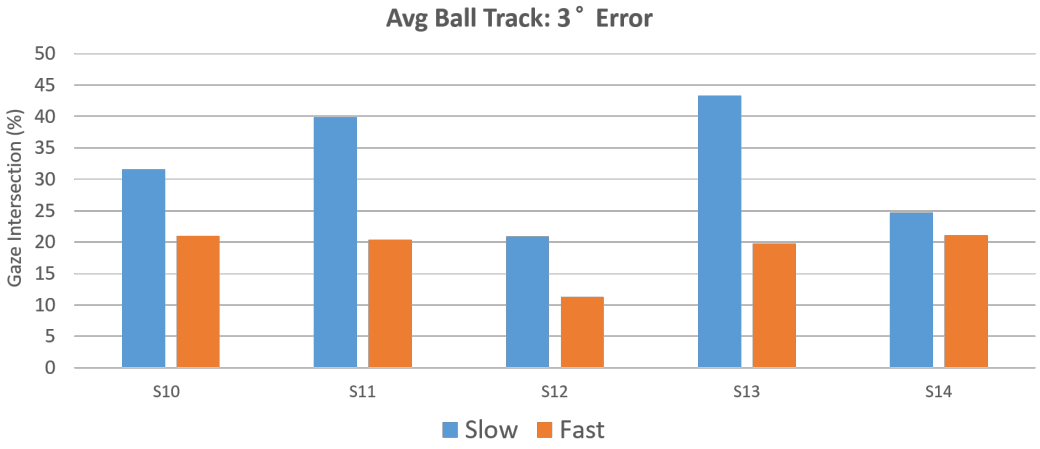

Figure 4.10: Chart of intersection of gaze with ball percentage for each subject when 3° of error correction was applied.

4.4 Eye and Head Movements

Since the experiment is reproducing the EIH velocities of a saccade it is important to analyze how much of the work was done by eyes and head when tracking the target. Given the naturalistic environment in VR we cannot fix their head position and the gaze in world signal then consists of both eye and head movements. Relatively the eyes are easier to move fast in relation to the head, but since they tend to overshoot with saccades head movements may be utilized to do a larger percentage of the work in some cases. In reality all trials feature some head movement, and tracking strategies can influence how much more work the eyes do than the head. This can be measured by computing the amplitude from the eye and head movements and generating the ratio of

eye to head amplitude at the end of the trial.

Using this ratio the strategies of eye and head movement coordination can be compared for the slow and fast phases of the experiment. Figure 4.11 shows a comparison of this ratio averaged across all trials for each subject. There are only 2 cases where the Eye/Head ratio was higher in the slow case than the fast phase. One of these was the aforementioned S12 whose track may not have been consistent for the entire experiment.

While watching subjects perform the experiment a hypothesis was formed that subjects tended to use more eye movements in the fast phase as it was necessary to successfully track and then shoot the ball. In the slow phase due to a longer delay before and after animation there is not much incentive to shoot the ball during motion, but to shoot it before or after it moves. Because of this a combination of eye and head movements are performed and the slow but steady head movements safely allows the subject to track the ball and score points while it is green. Overall the trend was not constant for all subjects with a higher ratio in the fast trial, but most of them had very close ratios. Only 3 subjects showed a large increase in eye/head ratio during the fast phase.

This suggests that while 3 subjects may have changed their strategy for tracking the ball to use more eye movements the conditions in these two phases were not enough to elicit uniformly different results across subjects. This analysis would benefit from filtering out trials with low gaze/ball intersection percentages, as trials where the ball is not actually tracked are contributing

to the average eye/head ratio.

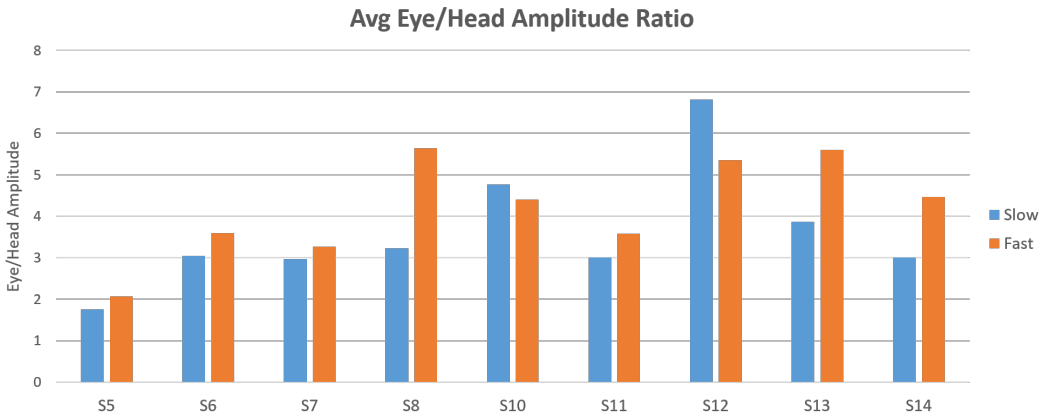

Figure 4.11: Chart of Eye/Head Amplitude ratio averaged over each subject. Averages were computed for both the slow and fast phases.

4.5 Eye Movement Reproduction

To conclude our analysis consider the original goal of the project, to elicit gaze behaviors similar to what was collected in the real world. For this experiment saccades were used and the velocity profile of gaze in world computed. Example comparisons of these profiles can be seen on the left of Figures 4.12 and 4.13. In these cases the gaze world does not accurately reproduce the ball velocity in blue.

Also evident in these Figures, specifically 4.13, the ball velocity profile itself seems noisy and not true to the saccade it was generated from. This could be due to data loss when converting the velocity to the approximating interpolating function F_c. Another potential cause is that under the hood the

animation function is not properly recreating the motion due to floating point precision, or the result of saccades with as few as 5 data samples which is 55ms and above the minimum duration of a saccade.

Overall, the goal of forcing subjects in a natural setting to perform specific gaze behaviors did not work when animating targets using saccadic velocities. A better experiment for eliciting saccades would to ensure the subject is facing directly forward, and to spawn a target with desired vertical and horizontal change drawn from the saccade data already in the final position. The subject would then shift attention to the target using saccade(s) which could then be compared to the real world data that generated it.

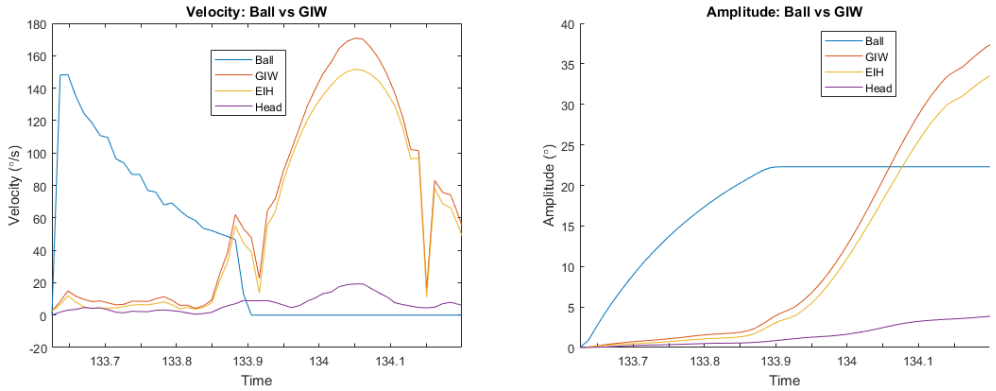

Figure 4.12: Pictured here is a case where eye movements dominated while tracking the ball. Left: Velocities of EIH, Ball, Head, and GIW over time. Right: Amplitudes of EIH, Ball, Head, and GIW over time.

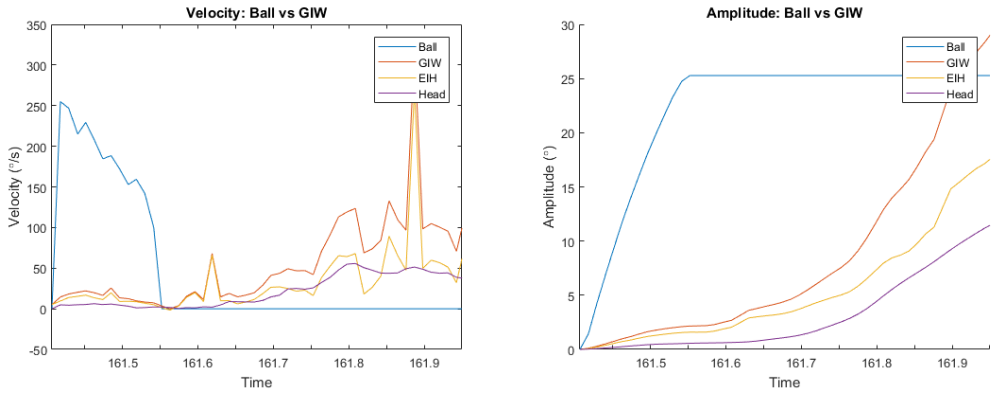

Figure 4.13: Pictured here is a case where both head movements and eye movements played a role in tracking the ball. Left: Velocities of EIH, Ball, Head, and GIW over time. Right: Amplitudes of EIH, Ball, Head, and GIW over time.

Chapter 5

Conclusion

Overall the goal of this project was to capture the characteristics of gaze behavior in the real world and use this to collect a similar dataset in VR. Specifically saccades and their defining characteristics (duration, peak velocity, and the velocity function itself) were targeted. This reproduction was not successful in the current dataset. As mentioned before, the coordination between eye and head movements is complicated and individual strategies play a role in how subjects will respond to the stimuli. The saccadic motions we produced replicated eye in head movements in a way that the head or eye could accomplish the task, and because of this the recorded gaze behavior was vastly different than the original data.

The notion of producing specific gaze behaviors without overtly forcing the subject to perform them is similar to the saying 'You can lead a horse to water but you can't make it drink.' While we can present all of the underlying cues needed to create the eye movement, we have no guarantee that the movement will be produced. Ultimately a system that achieves this goal can either reproduce specific gaze in world movements and process the data using post-hoc filters to see if it occurred, or in a more perfect solution recreate the

real world task in VR, and utilize subtle cues/events to force the subject to make similar but not exact movements. For example, with walking we can collect real world data, determine how many smooth pursuits occur and what objects created them. These cases can be considered and recreated in VR, with the hope that the subject tends to them in the same manner. The latter would take lots of work to create and implement in VR.

Overall, the data collected in this project is useful in that it provides insight into how eye and head movements are coordinated across different subjects. This is also a first attempt at recreating gaze behaviors in VR and lessons have been learned in how we can design future experiments.

Chapter 6

Future Work

A large contribution of this work is the gaze data collected and the system used to record and analyze said data. In this chapter future work to improve the accuracy of collected gaze data and how to improve on our current experiment will be elaborated.

6.1 Calibration/Accuracy Improvements

Eye tracking calibrations are extremely susceptible to error, and for this project the Pupil Labs VR eye tracker is used for the first time. In the future it would be best to implement a version of the calibration procedure within the Vizard code base itself instead of using the provided calibration in Unity. The API functions for this are clearly exposed in the provided example and can be used with the existing calibration procedure. One of the major flaws of the provided calibration routine is the calibration targets were presented outside the visible field of view within the VR headset. Also, targets on the outside of the field of view were susceptible to blurring due to the optics system and were only presented at one depth. Depending on the experiment an in house calibration could be tailored to produce data accurate at a variety of depths,

or a specific depth. Pairing this with the existing validation process would streamline the error analysis process. Also, for the future more than 9 points per plane should be used. The plots for Figures 4.2 to 4.4 can be misleading, as only 9 points are used 3 at each horizontal and 3 at each vertical value. Using more than 9 points to cover the same amount of the visual field will make it easier to confirm where error is lowest within the visual field.

Considering the current dataset a better process for calculating the ball and gaze intersection described in Section 4.3 could be implemented. Currently we pick one error value in degrees and apply it to the entire trial. Instead, for the subjects that performed the validation we can use the measured error and current gaze direction to determine how much error may be present at that point in time. This would produce more accurate results for the ball and gaze intersection analysis.

A correction procedure that can be applied to previously collected data and future data is to use the recorded error values from the validation and apply a correction to the gaze values provided by the eye tracker. This can be seen in Figure 6.1. A transform is generated by methods like Random Sampling Consensus (RANSAC) or a linear SVD that shifts and corrects the provided gaze data.

6.2 Experimental Updates

Given this was the first attempt at recreating real world gaze data in VR there are things that should be addressed before the next round of data

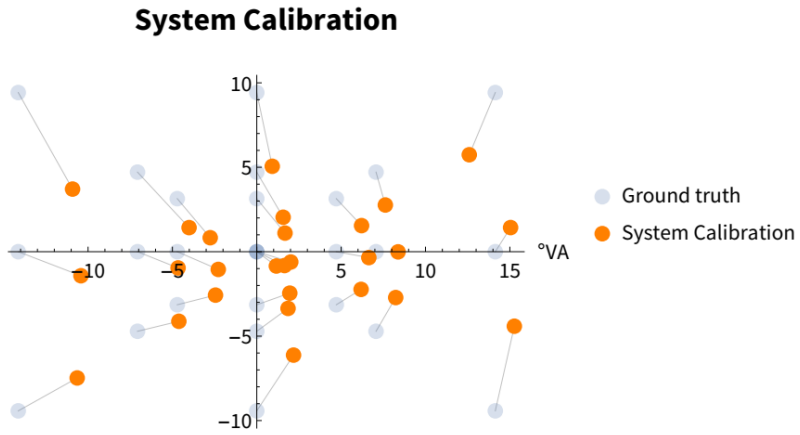

Figure 6.1: An illustration of applying a post-hoc gaze correction to gaze data. The validation procedure mentioned in Section 3.5 is performed and the recorded value is compared to the ground truth point. To correct this error a linear transform is generated that maps the system values in orange to the ground truth values in blue. The transform generated minimizes the error between all ground truth points and their corresponding data points. This transform is applied to future data to produce a more accurate gaze data stream.

collection.

First, the way targets were animated in the current experiment was a linear path from their starting to point to the ending point. This would cause trials that had large values for ϕ or θ to essentially fly by the user's head. This can be visualized in Figure 6.2. Instead of using a linear path this problem can be addressed by animating the target along a curve around the user's head. This can implemented my providing the head position as a pivot point so the animating function generates a path by rotating the target around the user's head. Using this method would ensure the target will always remain at

a uniform distance from the head's starting position, allowing the stimuli to cover a more consistent portion of the user's field of view.

Figure 6.2: Top down view of how targets are animated around the subject during experiment. Green indicates the starting point in front of the user, and red indicates the end position. Pictured on the left is the current implementation, where the target is animated along a straight line between then points. In the future the animation should happen as shown on the right, where the target is rotated around the subject as it is animated towards its destination.

Also, there are currently no constraints or filtering applied to the real world saccades. In the future the most extraneous of these (covering most distance) are not well reproduced. Filtering these outlier cases would prevent the user from making large movements with high ball tracking error, and allow the user to process targets more consistently over time.

Next, with the current experiment an eye tracking calibration and validation are performed only at the beginning of the experiment. With the move to longer data collection sessions it would be best to mix the calibration/validation procedure into the experiment as well. The current experiment code allows for a calibration to be performed after a configurable number of trials have been performed. This would ensure that the currently generated gaze data is as accurate as can be and allows us to see how gaze error drifts

over time.

Once these experiment specific changes have been made the next goal is to collect a larger dataset where each subject has the exact same data and environment, unlike the current dataset. In the future longer experiments can be run, and money used as an incentive to keep subjects interested for longer. As mentioned earlier eye tracking data depends on eye characteristics and spreading data collection over more demographics like age, race, and gender would create a more well rounded data analysis, and aid in any machine learning efforts the resulting data is utilized for.

To benefit data analysis the experiment should also incorporate a measure of whether the intended gaze behavior was reproduced or not, for any elicited behaviors. This allows the hypothesis of whether the gaze movements were recreated to be tested concretely. For example, with the current dataset one method for testing the reproduction of a velocity profile is to align the gaze in world velocity with the Gaussian like saccade velocity curve (Figure 3.3) and compute the error in overlapping area. This would be 0 in the perfect case, which would likely never happen, and be higher in case where the velocity profile does not behave similar to the input data.

Lastly, while future data may tend to be more faithful to natural statistics it is limited by the tasks performed during data collection. Naturally the future tasks for VR should begin to mirror the real world data collection environment, with less constraints and rules for any of the game based tasks. This could be accomplished by having them perform the same task in a vi-

sual similar environment and analyzing the difference between these two gaze distributions.

Bibliography

[1] N Anantrasirichai, Iain D Gilchrist, and David R Bull. Fixation identification for low-sample-rate mobile eye trackers. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3126–3130. IEEE, 2016.

[2] Richard Andersson, Linnea Larsson, Kenneth Holmqvist, Martin Stridh, and Marcus Nyström. One algorithm to rule them all? an evaluation and discussion of ten eye movement event-detection algorithms. *Behavior research methods*, 49(2):616–637, 2017.

[3] Guy Thomas Buswell. How people look at pictures: a study of the psychology and perception in art. 1935.

[4] Roger HS Carpenter. *Movements of the Eyes, 2nd Rev.* Pion Limited, 1988.

[5] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[6] Han Collewijn, Robert M Steinman, Casper J Erkelens, Zygmunt Pizlo, Eileen Kowler, and Johannes Van der Steen. Binocular gaze control under free-head conditions. In *Vestibular and brain stem control of eye, head and body movements*, pages 203–220. Karger Publishers, 1992.

[7] Han Collewijn, Robert M Steinman, Casper J Erkelens, Zygmunt Pizlo, and Johannes van der Steen. Effect of freeing the head on eye movement characteristics during three-dimensional shifts of gaze and tracking. *The head-neck sensory motor system. Oxford University Press, Oxford*, pages 412–418, 1992.

[8] Gabriel Diaz, Kamran Binaee, and Flip Phillips. Predictive movements of the hands and eyes to a target that disappears briefly when moving in depth. *Journal of Vision*, 16(12):1349–1349, 2016.

[9] Gabriel Diaz, Joseph Cooper, Dmitry Kit, and Mary Hayhoe. Real-time recording and classification of eye movements in an immersive virtual environment. *Journal of vision*, 13(12):5–5, 2013.

[10] David L Donoho. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627, 1995.

[11] Andrew Duchowski, Eric Medlin, Nathan Cournia, Hunter Murphy, Anand Gramopadhye, Santosh Nair, Jeenal Vorah, and Brian Melloy. 3-d eye movement analysis. *Behavior Research Methods*, 34(4):573–591, 2002.

[12] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[13] Mary Hayhoe and Dana Ballard. Eye movements in natural behavior. *Trends in cognitive sciences*, 9(4):188–194, 2005.

[14] Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost Van de Weijer. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011.

[15] Sabrina Hoppe and Andreas Bulling. End-to-end eye movement detection using convolutional neural networks. *arXiv preprint arXiv:1609.02452*, 2016.

[16] Moritz Kassner, William Patera, and Andreas Bulling. Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14 Adjunct, pages 1151–1160, New York, NY, USA, 2014. ACM.

[17] Thomas Kinsman, Karen Evans, Glenn Sweeney, Tommy Keane, and Jeff Pelz. Ego-motion compensation improves fixation detection in wearable eye tracking. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 221–224. ACM, 2012.

[18] Christopher M Knapp, Irene Gottlob, Rebecca J McLean, and Frank A Proudlock. Horizontal and vertical look and stare optokinetic nystagmus symmetry in healthy adult volunteers. *Investigative ophthalmology & visual science*, 49(2):581–588, 2008.

[19] Michael F Land. The coordination of rotations of the eyes, head and trunk in saccadic turns produced in natural situations. *Experimental brain research*, 159(2):151–160, 2004.

[20] Otto Lappi. Eye movements in the wild: oculomotor control, gaze behavior & frames of reference. *Neuroscience & Biobehavioral Reviews*, 69:49–68, 2016.

[21] Linnéa Larsson, Marcus Nyström, Richard Andersson, and Martin Stridh. Detection of fixations and smooth pursuit movements in high-speed eye-tracking data. *Biomedical Signal Processing and Control*, 18:145–152, 2015.

[22] Linnéa Larsson, Marcus Nyström, Håkan Ardö, Kalle Åström, and Martin Stridh. Smooth pursuit detection in binocular eye-tracking data with automatic video-based performance evaluation. *Journal of Vision*, 16(15):20–20, 2016.

[23] Linnéa Larsson, Marcus Nyström, and Martin Stridh. Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit. *IEEE Transactions on Biomedical Engineering*, 60(9):2484–2493, 2013.

[24] Linnéa Larsson, Andrea Schwaller, Marcus Nyström, and Martin Stridh. Head movement compensation and multi-modal event detection in eye-tracking data for unconstrained head movements. *Journal of Neuroscience Methods*, 274:13–26, 2016.

[25] Susana Martinez-Conde, Stephen L Macknik, and David H Hubel. The role of fixational eye movements in visual perception. *Nature Reviews Neuroscience*, 5(3):229–240, 2004.

[26] Susana Martinez-Conde, Stephen L Macknik, Xoana G Troncoso, and Thomas A Dyar. Microsaccades counteract visual fading during fixation. *Neuron*, 49(2):297–305, 2006.

[27] Susana Martinez-Conde, Stephen L Macknik, Xoana G Troncoso, and David H Hubel. Microsaccades: a neurophysiological analysis. *Trends in neurosciences*, 32(9):463–475, 2009.

[28] Marcus Nyström and Kenneth Holmqvist. An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior research methods*, 42(1):188–204, 2010.

[29] Marcus Nyström, Ignace Hooge, and Kenneth Holmqvist. Post-saccadic oscillations in eye movement data recorded with pupil-based eye trackers reflect motion of the pupil inside the iris. *Vision research*, 92:59–66, 2013.

[30] Pontus Olsson. Real-time and offline filters for eye tracking, 2007.

[31] Justus J Randolph. Free-marginal multirater kappa (multirater k [free]): An alternative to fleiss' fixed-marginal multirater kappa. *Online submission*, 2005.

[32] Martin Rolfs. Microsaccades: small steps on a long way. *Vision research*, 49(20):2415–2441, 2009.

[33] Dario D Salvucci and John R Anderson. Tracing eye movement protocols with cognitive process models. 1998.

[34] Dario D Salvucci and Joseph H Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, pages 71–78. ACM, 2000.

[35] Robert M Steinman, Eileen Kowler, and Han Collewijn. New directions for oculomotor research. *Vision research*, 30(11):1845–1864, 1990.

[36] William Tasman and Edward A Jaeger. *Duane's Ophthalmology*. Lippincott, Williams & Wilkins, 2009.

[37] Benjamin W Tatler, James R Brockmole, and RHS Carpenter. Latest: A model of saccadic decisions in space and time. *Psychological Review*, 124(3):267, 2017.

[38] Matteo Tomasi, Shrinivas Pundlik, Alex R Bowers, Eli Peli, and Gang Luo. Mobile gaze tracking system for outdoor walking behavioral studies. *Journal of vision*, 16(3):27–27, 2016.

[39] Mélodie Vidal, Andreas Bulling, and Hans Gellersen. Detection of smooth pursuits using eye movement shape features. In *Proceedings of the symposium on eye tracking research and applications*, pages 177–180. ACM, 2012.

[40] Heino Widdel. Operational problems in analysing eye movements. *Advances in psychology*, 22:21–29, 1984.

[41] Wikipedia, the free encyclopedia. Human photoreceptor distribution, 2013. [Online; accessed June 12, 2017].

[42] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 131–138. ACM, 2016.

[43] Alfred L Yarbus. *Eye movements during perception of complex objects*. Springer, 1967.

[44] Raimondas Zemblys. Eye-movement event detection meets machine learning. *BIOMEDICAL ENGINEERING 2016*, 20(1), 2017.

[45] Raimondas Zemblys, Diederick C Niehorster, Oleg Komogortsev, and Kenneth Holmqvist. Using machine learning to detect events in eye-tracking data. *Behavior Research Methods*, pages 1–22, 2017.

Vita

Brendan John was born on August 2, 1993 to Dr. Randy John and Julie Zoccali-John of Salamanca, NY. He is currently pursuing his BS/MS at the Rochester Institute of Technology in Rochester, NY. His BS degree is in Computational Mathematics, and his MS degree is in Computer Science. His research interests include Computer Graphics, Computer Vision, and various fields of Mathematics with a specialization in eye tracking technology and human vision.

Permanent address: 73 Fawn Ave.
Salamanca, New York 14779

This thesis was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.